PROTOTIPO DE OSCILOSCOPIO DIGITAL BASADO EN DSP CON LCD GRÁFICA

AUTORES:
JULIAN EDUARDO BADILLO HERNÁNDEZ
ANYELA CATERINE CABRA BRIJALDO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
BUCARAMANGA
2005

PROTOTIPO DE OSCILOSCOPIO DIGITAL BASADO EN DSP CON LCD GRÁFICA

AUTORES:
JULIAN EDUARDO BADILLO HERNÁNDEZ
ANYELA CATERINE CABRA BRIJALDO

PROYECTO DE GRADO

DIRECTORES:
MsC. JAIME GUILLERMO BARRERO PÉREZ
Ing. JOSÉ ALEJANDRO AMAYA PALACIO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS
ESCUELA DE INGENIERÍAS ELÉCTRICA, ELECTRÓNICA Y TELECOMUNICACIONES
BUCARAMANGA
2005

UNIVERSIDAD
INDUSTRIAL DE
SANTANDER

NOTA DEL PROYECTO DE GRADO

Nombre del Estudiante		CÓDIGO
JULIÁN EDUARDO BADILLO HERNÁNDEZ		1982218
TITULO DEL PROYECTO		
PROTOTIPO DE OSCILOSCOPIO DIGITAL BASADO EN DSP CON LCD GRÁFICA		
CODIGO	FACULTAD	CARRERA
01200507	INGENIERÍAS FISICOMECÁNICAS	INGENIERÍA ELECTRÓNICA
CALIFICACIÓN		CRÉDITOS
CUATRO, SIETE (4.7)		15

DIRECTOR DEL PROYECTO

NOMBRE	FIRMA
JAIME GUILLERMO BARRERO PÉREZ-DIRECTOR	JOSÉ ALEJANDRO AMAYA PALACIO-CODIRECTOR

CALIFICADORES

		FECHA		
		A	M	D
F CÉSAR ANTONIO DUARTE GUALDRÓN	F LINA MARCELA FUENTES VILLA	2005	09	16

UNIVERSIDAD
INDUSTRIAL DE
SANTANDER

NOTA DEL PROYECTO DE GRADO

NOMBRE DEL ESTUDIANTE		CÓDIGO
JULIÁN EDUARDO BADILLO HERNÁNDEZ		1982218
TITULO DEL PROYECTO		
PROTOTIPO DE OSCILOSCOPIO DIGITAL BASADO EN DSP CON LCD GRÁFICA		
REGISTRO No.	FACULTAD	CARRERA
01200507	INGENIERÍAS FISICOMECÁNICAS	INGENIERÍA ELECTRÓNICA
CALIFICACIÓN		CRÉDITOS
APROBADO		15

DIRECTOR DEL PROYECTO

NOMBRE	FIRMA
JAIME GUILLERMO BARRERO PÉREZ-DIRECTOR	JOSÉ ALEJANDRO AMAYA PALACIO-CODIRECTOR

CALIFICADORES

		FECHA		
		A	M	D
F N CÉSAR ANTONIO DUARTE GUALDRÓN	F N LINA MARCELA FUENTES VILLA	2005	09	16

UNIVERSIDAD INDUSTRIAL DE SANTANDER

NOTA DEL PROYECTO DE GRADO

Nombre del Estudiante		CÓDIGO
ANYELA CATERINE CABRA BRIJALDO		1996220
TITULO DEL PROYECTO		
PROTOTIPO DE OSCILOSCOPIO DIGITAL BASADO EN DSP CON LCD GRÁFICA		
CODIGO	FACULTAD	CARRERA
01200507	INGENIERÍAS FISICOMECÁNICAS	INGENIERÍA ELECTRÓNICA
CALIFICACIÓN CUATRO, SIETE (4.7)		CRÉDITOS 15

DIRECTOR DEL PROYECTO

NOMBRE	FIRMA
JAIME GUILLERMO BARRERO PEREZ- DIRECTOR	JOSÉ ALEJANDRO AMAYA PALACIO-CODIRECTOR

CALIFICADORES

		FECHA A	M	D
F CÉSAR ANTONIO DUARTE GUALDRÓN	F LINA MARCELA FUENTES VILLA	2005	09	16

UNIVERSIDAD INDUSTRIAL DE SANTANDER

NOTA DEL PROYECTO DE GRADO

NOMBRE DEL ESTUDIANTE		CÓDIGO
ANYELA CATERINE CABRA BRIJALDO		1996220
TITULO DEL PROYECTO		
PROTOTIPO DE OSCILOSCOPIO DIGITAL BASADO EN DSP CON LCD GRÁFICA		
REGISTRO No.	FACULTAD	CARRERA
01200507	INGENIERÍAS FISICOMECÁNICAS	INGENIERÍA ELECTRÓNICA
CALIFICACIÓN APROBADO		CRÉDITOS 15

DIRECTOR DEL PROYECTO

NOMBRE	FIRMA
JAIME GUILLERMO BARRERO PEREZ-DIRECTOR	JOSÉ ALEJANDRO AMAYA PALACIO-CODIRECTOR

CALIFICADORES

		FECHA A	M	D
F N CÉSAR ANTONIO DUARTE GUALDRÓN	F N LINA MARCELA FUENTES VILLA	2005	09	16

DEDICATORIA

A mis padres, Alvaro y Cecilia,
Como una pequeña retribución a su gran amor y sacrificio.

A mis hermanas, Diana Y Sandra

A mi tía Carmen

A Zahira

Julian

A mis padres, Belarmino y María Antonia
Por todo su amor y esfuerzo.

A mi hermano, Stiven
Por su cariño y colaboración.

A Fernando y Virginia
Porque su invaluable ayuda y confianza.

A Beto
Por todo su afecto, paciencia y comprensión.

Caterine

AGRADECIMIENTOS

Los autores desean agradecerle, en primer lugar a Dios por permitirnos culminar esta etapa.

A nuestros directores de proyecto, Jaime Guillermo Barrero P. y José Alejandro Amaya P. por su orientación en la realización de este proyecto. A los profesores Cesar Duarte, David Miranda, Javier Mier y Jairo Mantilla por su valiosa asesoría.

A nuestros familiares, amigos y compañeros por su apoyo y colaboración a lo largo de toda nuestra carrera universitaria.

TABLA DE CONTENIDO

INTRODUCCIÓN 1

1. FUNDAMENTOS TEÓRICOS 3
1.1. DEFINICIONES Y TÉRMINOS 3
1.1.1. Tiempo (t) 3
1.1.2. Magnitud (m) 3
1.1.3. Onda 4
1.1.4. Pulso 4
1.1.5. Forma de onda 4
1.1.6. Periodicidad (Aperiodicidad) 4
1.1.7. Periodo 4
1.1.8. Frecuencia 4
1.1.9. Pico positivo (negativo) 5
1.1.10. Valor pico a pico 5
1.1.11. Raíz media cuadrática (RMS) 5
1.1.12. Valor promedio 5
1.2. PROCESADOR DE SEÑALES DIGITALES 6
1.2.1. Características del DSP56F8323 7
1.2.2. Circuitos periféricos 8

2. HARDWARE 10
2.1. TARJETA DE DESARROLLO 56F8300 10
2.1.1. Especificaciones 11
2.1.2. Opciones 12
2.1.3. Puertos 13
2.1.4. Indicadores 20

2.2. PANTALLA GRÁFICA LCD 22
2.3. PROTOTIPO FINAL 32
2.3.1. ESPECIFICACIONES 34
2.3.2. ACONDICIONAMIENTO DE SEÑALES 37

3. SOFTWARE 45
3.1. CONFIGURACIÓN DEL DSP 45
3.1.1. Conversor Analógico / Digital (ADC) 45
3.1.2. Timer (TMR) 53
3.1.3. Pines de propósito general (GPIO) 53
3.2. FILTRO DIGITAL 59
3.3. MEDICIÓN DE LOS PARÁMETROS DE LA SEÑAL 64
3.3.1. Cálculo del periodo 64
3.3.2. Cálculo del valor promedio 67
3.3.3. Cálculo del valor RMS 68
3.3.4. Cálculo del valor pico a pico 70
3.4. INICIALIZACIÓN Y MANEJO DE LA PANTALLA GRÁFICA LCD 71

4. PRUEBAS Y RESULTADOS OBTENIDOS 74
4.1. PRUEBAS EN LA PANTALLA GRÁFICA HG25504NG-01 DE HYUNDAI. 74
4.2. PRUEBAS SOBRE LAS MEDICIONES DE PARÁMETROS CON UNA SEÑAL DE VOLTAJE PICO A PICO CONSTANTE. 75
4.3. PRUEBAS SOBRE LAS MEDICIONES DE PARÁMETROS CON UNA SEÑAL DE FRECUENCIA CONSTANTE 83
4.3. PRUEBAS SOBRE LAS MEDICIONES DE PARÁMETROS MÁXIMOS Y MÍNIMOS EN UNA SEÑAL DE ENTRADA. 89
4.4. PRUEBAS SOBRE LA VISUALIZACIÓN DE LA FORMA DE ONDA DE UNA SEÑAL DE ENTRADA. 89

5. CONCLUSIONES Y RECOMENDACIONES 92

6. REFERENCIAS BIBLIOGRÁFICAS..... 95

7. BIBLIOGRAFÍA..... 96

ANEXO A. CODIGO GENERAL DEL PROGRAMA..... 98
ANEXO B. PROGRAMA DEL FILTRO DIGITAL EN MATLAB. 128
ANEXO C. PROGRAMA DEL FILTRO DIGITAL EN CODEWARRIOR..... 129
ANEXO D. MANUAL DE USUARIO..... 134

LISTA DE FIGURAS

Figura 1. Diagrama de pines del DSP 56F8323 7
Figura 2. Tarjeta de desarrollo 56F8300 11
Figura 3. Esquema de alimentación de la tarjeta de desarrollo 56F8300 12
Figura 4. Esquemático Tarjeta de Desarrollo 56F8300 14
Figura 5. Puerto ADC 15
Figura 6. Puerto GPIO / SERIAL 15
Figura 7. Puerto TIMER / PWM 16
Figura 8. Puerto SCI1 17
Figura 9. Esquema de conexión de la interfase serial 17
Figura 10. Puerto JTAG / EOnCE 18
Figura 11. Puerto P1 – Host JTAG 19
Figura 12. Esquema de conexión de la interfase de programación 20
Figura 13. Esquema de conexión de los Indicadores 21
Figura 14. LCD Gráfica HG25504NG-01 de Hyundai 23
Figura 15. Diagrama de bloques de la LCD 24
Figura 16. Distribución de la pantalla de texto 25
Figura 17. Generador interno de caracteres 25
Figura 18. Distribución de la pantalla gráfica 26
Figura 19. Diagrama de conexión de la LCD. 27
Figura 20. Prototipo final *DIGISCOPE*. 32
Figura 21. Diagrama de bloques canal del osciloscopio. 33
Figura 22. Inicialización incorrecta de la pantalla LCD gráfica. 34
Figura 23. Tarjeta de Propósito Específico *DIGISCOPE*. 35
Figura 24. Esquema de alimentación del prototipo *DIGISCOPE* 35
Figura 25. Interfase de programación en el prototipo final 36

Figura 26. Circuito de atenuación y ajuste 39
Figura 27. Análisis espectral de la señal analógica $x_a(t)$ 41
Figura 28. Respuesta en frecuencia del circuito integrado LTC1563-2 42
Figura 29. Circuito utilizado para la implementación del filtro 43
Figura 30. Propiedades del *Bean* ADC 46
Figura 31. Registro *ADC Control Register 1* (ADCR1) 47
Figura 32. Registro *ADC Control Register 2* (ADCR2) 48
Figura 33. *Registro ADC Zero Crossing Control Register* (ADZCC) 48
Figura 34. Registro *ADC Sample Disable Register* (ADSDIS) 49
Figura 35. Registro *ADC Status Register* (ADSTAT) 50
Figura 36. Registros *ADC Result Registers* (ADRSLT0–7) 50
Figura 37. Registros *ADC Offset Registers* (ADOFS0–7) 51
Figura 38. Diagrama de Flujo de Adquisición de la Señal 52
Figura 39. Propiedades del *Bean FreeCntr16* 53
Figura 40. Registro *Data Register* (DR) 55
Figura 41. Registro Data Direction Register (DDR) 56
Figura 42. Registro *Peripheral Enable Register* (PER) 56
Figura 43. Propiedades del *Bean BitIO* 57
Figura 44. Diagrama de flujo de configuración de un puerto como GPIO 58
Figura 45. Respuesta en frecuencia del filtro FIR pasabajas diseñado. 63
Figura 46. Detalle de la respuesta en frecuencia del filtro FIR. 63
Figura 47. Diagrama de flujo para el cálculo del periodo. 66
Figura 48. Diagrama de flujo del cálculo del valor promedio 68
Figura 49. Diagrama de flujo del cálculo del valor RMS 69
Figura 50. Diagrama de flujo del cálculo del valor pico a pico 70
Figura 51. Diagrama de flujo inicialización LCD 71
Figura 52. Diagrama de flujo escritura de datos en la LCD 72
Figura 53. Diagrama de flujo código ASCII de variables 73
Figura 54. Portada del prototipo *DIGISCOPE.* 74
Figura 55. Visualización del marco en la pantalla LCD 75

Figura 56. Error en la medición de valores de frecuencia. 80
Figura 57. Error en la medición de valores de voltaje pico a pico. 81
Figura 58. Error en la medición de valores de voltaje RMS. 82
Figura 59. Error en las mediciones del voltaje pico a pico con frecuencia de 10kHz..... 86
Figura 60. Error en las mediciones del voltaje RMS con frecuencia de 10kHz. 87
Figura 61. Error en las mediciones de frecuencia debido a la variación de voltaje pico a pico. 88
Figura 62. Visualización de la forma de onda de una señal senoidal a 60Hz. 90
Figura 63. Visualización de la forma de onda de una señal triangular. 90
Figura 64. Visualización de la forma de onda de una señal cuadrada a 1,2kHz..... 91
Figura 65. Visualización de la forma de onda de una señal cuadrada a 5,4kHz..... 91

LISTA DE TABLAS

Tabla 1. Configuración de los puertos GPIO en el DSP 56F8323 9
Tabla 2. Indicadores Tarjeta de Desarrollo 56F8300 21
Tabla 3. Características de la LCD HG25504NG-01 de Hyundai 22
Tabla 4. Descripción de los pines de la LCD 28
Tabla 5. Configuración de los pines GPIO para el manejo de la LCD 29
Tabla 6. Descripción de comandos del controlador gráfico 29
Tabla 7. Funciones de los botones del prototipo final. 33
Tabla 8. Pines GPIO utilizados para controlar los *switches* análogos 39
Tabla 9. Pines GPIO en el DSP 56F8323 54
Tabla 10. Configuración de los pines GPIO en el prototipo 58
Tabla 11. Mediciones de frecuencia con un voltaje pico a pico de 2V. 76
Tabla 12. Mediciones de voltaje pico a pico con respecto a la variación de frecuencia. 77
Tabla 13. Mediciones de valor promedio con respecto a la variación de frecuencia. 77
Tabla 14. Mediciones de valor RMS con respecto a la variación de frecuencia. 78
Tabla 15. Porcentajes de error en la medición de valores de frecuencia..... 79
Tabla 16. Porcentajes de error en la medición de valores de voltaje pico a pico. 80
Tabla 17. Porcentajes de error en la medición de valores de Voltaje RMS. 81
Tabla 18. Mediciones de voltaje pico a pico con frecuencia de 10kHz. 83
Tabla 19. Mediciones de voltaje promedio con frecuencia de 10kHz, ... 84
Tabla 20. Mediciones de voltaje RMS con frecuencia de 10kHz. 84

Tabla 21. Mediciones de frecuencia con respecto al voltaje medido en el TEKTRONIX. 85
Tabla 22. Porcentaje de error en las mediciones de Voltaje pico a pico con frecuencia de 10KHz. 85
Tabla 23. Porcentaje de error en las mediciones de voltaje RMS con frecuencia de 10kHz. 87
Tabla 24. Porcentaje de error en las mediciones de frecuencia debidas a la variación de voltaje pico a pico. 88

RESUMEN

TÍTULO: PROTOTIPO DE OSCILOSCOPIO DIGITAL BASADO EN DSP CON LCD GRÁFICA.

AUTORES: BADILLO HERNANDEZ, JULIAN EDUARDO, y, CABRA BRIJALDO, ANYELA CATERINE**

PALABRAS CLAVES: Osciloscopio, frecuencia, valor pico a pico, valor RMS, valor Promedio, DSP y LCD gráfica.

DESCRIPCIÓN:

En el documento se describe la realización de una tarjeta de propósito general 56F8300 y el diseño e implementación de un prototipo de osciloscopio digital de un canal basado en el DSP 56F8323 con pantalla LCD gráfica que permite medir los parámetros fundamentales de señales de voltaje adquiridas como son frecuencia, valor pico a pico, valor RMS y valor promedio.

Para la digitalización de las señales se utilizó el Conversor Analógico/Digital del DSP 56F8323 de Motorola y la plataforma de desarrollo CodeWarrior con la cual se implementó el algoritmo para la adquisición, la medición de los parámetros y la visualización de la señales en la pantalla LCD gráfica.

En el primer capítulo se hace una descripción de la terminología utilizada en el trabajo con digitalizadores de formas de onda y una descripción general del DSP 56F8323 de Motorola.

El hardware utilizado e implementado en la tarjeta de propósito general 56F8300 y en el prototipo final *DIGISCOPE* se expone en el segundo capítulo. En el tercer capítulo, se explican los algoritmos desarrollados para la adquisición, procesamiento y visualización de la señales.

En el cuarto capítulo se presentan los resultados de las pruebas realizadas con el fin de comparar la forma de onda visualizada y los parámetros medidos con los osciloscopios digitales TEKTRONIX TDS 210 y FLUKE 105 Series II disponibles en los laboratorios de la Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones.

Los algoritmos en la medición de los parámetros demostraron ser adecuados para la aplicación ya que en las pruebas realizadas al prototipo se obtuvieron porcentajes de error bajos al compararlos con los otros dos dispositivos.

Finalmente en el último capítulo se menciona las conclusiones y recomendaciones.

---

* Trabajo de Grado.

** Facultad de Ingenierías Físico-mecánicas. Ingeniería Electrónica. Director: MsC. Jaime Guillermo Barrero Pérez.

SUMMARY

TITLE: PROTOTYPE OF A DIGITAL OSCILLOSCOPE BASED ON DSP WITH GRAPHIC LCD.

AUTHORS: BADILLO HERNANDEZ, JULIAN EDUARDO, and, CABRA BRIJALDO, ANYELA CATERINE**

KEY WORDS: Oscilloscope, frequency, peak to peak value, RMS value, average value, DSP and graphic LCD.

DESCRIPTION:

This document describes the implementation of a development board for the DSP 56F8323 from Motorola and the design and implementation of a prototype of a digital oscilloscope based on the DSP 56F8323 with graphic LCD that allows the measurement of the fundamental parameters (frequency, peak to peak value, RMS value and average value) of the sampled voltage signals.

The analog to digital converter embedded in the DSP 56F8323 and the CodeWarrior platform were used for both, the digitalization of the signals and the design of the algorithms for the acquisition, parameter measurement and the visualization of the waveforms on the graphic LCD.

The first chapter of this document presents some of the terms and definitions used on the work with digitizing waveform recorders and a general description of the DSP 56F8323 from Motorola.

The hardware designed and implemented for the development board and the final prototype, DIGISCOPE, are described in detail in the second chapter. The third chapter reviews the algorithms designed for the acquisition, processing and visualization of the waveforms.

The fourth chapter presents the various tests carried out to determine the performance of the prototype DIGISCOPE, in comparison with the other two digital oscilloscopes available in the school laboratories (TEKTRONIX TDS 210 and FLUKE 105 Series II) as well as the results obtained through these tests.

The algorithms designed for the measurement of the different parameters proved to be accurate, since the tests performed delivered small error percentages in comparison with the measurements of the other two devices.

Finally, the conclusions and recommendations are mentioned in the final chapter.

---

* Degree Work.

** Faculty of Physical-Mechanical Engineering. Electronics Engineering. Director: MsC. Jaime Guillermo Barrero Pérez.

INTRODUCCIÓN

Un osciloscopio es básicamente un sistema que realiza representaciones gráficas de señales eléctricas variables en el tiempo.

Los equipos electrónicos se dividen en dos clases: analógicos y digitales. Los primeros trabajan con variables continuas, mientras que los segundos lo hacen con variables discretas. Los osciloscopios analógicos funcionan aplicando directamente la tensión medida a unas placas de desviación dentro de un tubo de rayos catódicos. La tensión desvía el haz de electrones en sentido vertical proporcionalmente a su valor, trazando la forma de onda en la pantalla. A diferencia de estos, los osciloscopios digitales muestrean la forma de onda utilizando un conversor analógico/digital (ADC) para almacenar digitalmente la señal de entrada. Luego utilizan esta información para reconstruir la forma de onda en una pantalla.

Desde su desarrollo, los DSPs (*Digital Signal Processors*) se han convertido en parte integral de los sistemas de adquisición de señales no sólo por su velocidad de medición y cálculo sino por las nuevas posibilidades que han generado en el trabajo con estos sistemas.

El principal objetivo de este proyecto es implementar una tarjeta de desarrollo de propósito digital para el DPS 56F8323 de Motorota que permita realizar las pruebas preliminares al software de adquisición, procesamiento y visualización de las señales, y, construir un prototipo de un Osciloscopio Digital de un canal basado en DSP con pantalla LCD gráfica que permita medir los parámetros fundamentales de las señales de voltaje adquiridas.

Para la digitalización de las señales se utilizó el Conversor Analógico/Digital del DSP 56F8323 de Motorola y la plataforma de desarrollo CodeWarrior con la cual se implementó el algoritmo para la adquisición, la medición de los parámetros (frecuencia, valor medio, valor RMS, voltaje pico a pico) y la visualización de la señales en la pantalla LCD gráfica.

El prototipo se limitó a un solo canal debido a los altos costos de los dispositivos utilizados en la implementación del canal (etapas de acople, atenuación y filtrado) y la necesidad de importar los circuitos integrados[1].

En el primer capítulo se hace una descripción de la terminología utilizada en el trabajo con digitalizadores de formas de onda y una descripción general del DSP 56F8323 de Motorola.

El hardware utilizado e implementado en el desarrollo de este proyecto se describe en el segundo capítulo. En el tercer capítulo, se explican los algoritmos desarrollados para la adquisición, procesamiento y visualización de la señales.

Finalmente en los capítulos 4 y 5 se exponen los resultados de las pruebas realizadas al prototipo y las conclusiones, respectivamente.

[1] El costo de la implementación de un canal es $150000 aproximadamente.

1. FUNDAMENTOS TEÓRICOS

En este capítulo se exponen las principales definiciones de la terminología utilizada en el trabajo con digitalizadores de formas de onda, además de una descripción general del DSP 56F8323 de Motorola.

1.1. DEFINICIONES Y TÉRMINOS

Las definiciones que aquí se presentan son aplicables en general a la tecnología de pulsos en el dominio del tiempo de acuerdo al estándar *IEEE Standard Pulse Terms and Definitions 194-1977* de *The Institute of Electrical and Electronics Engineers, Inc.* y se han tomado directamente de este documento.

A lo largo de este documento se asume un sistema de coordenadas Cartesiano rectangular.

1.1.1. Tiempo (t)

Es la variable independiente tomada a lo largo del eje horizontal, incrementando en el sentido positivo de izquierda a derecha.

1.1.2. Magnitud (m)

Es la variable dependiente tomada a lo largo del eje vertical, incrementado en el sentido o polaridad positiva de abajo hacia arriba.

1.1.3. Onda

Una modificación del estado físico de un medio que se propaga en el mismo en función del tiempo y como resultado de una o más perturbaciones.

1.1.4. Pulso

Una onda que parte de un primer estado nominal, alcanza un segundo estado nominal, y finalmente retorna al primer estado nominal. El término pulso está incluido dentro del término onda.

1.1.5. Forma de onda

Una manifestación o representación (ya sea una, gráfica, traza, presentación de un osciloscopio, ecuación(es), tabla de datos coordinados o estadísticos, etc.) o una visualización de una onda, pulso o transición.

1.1.6. Periodicidad (Aperiodicidad)

Relacionado a una serie de formas de onda o rasgos que se repiten o reaparecen regularmente (irregularmente) en el tiempo.

1.1.7. Periodo

El valor absoluto del intervalo mínimo después del cuál reaparecen las mismas características de una forma de onda periódica o rasgo periódico.

1.1.8. Frecuencia

El recíproco del periodo.

1.1.9. Pico positivo (negativo)

La magnitud máxima (mínima).

1.1.10. Valor pico a pico

El valor absoluto de la diferencia algebraica entre la magnitud del pico positivo y la magnitud del pico negativo.

1.1.11. Raíz media cuadrática (RMS)

La raíz cuadrada del promedio del cuadrado de la magnitud. Si la magnitud está representada por n valores discretos m_j, la magnitud de la raíz media cuadrática es

$$M_{rms} = \left[\left(\frac{1}{n} \right) \sum_{j=1}^{j=n} \left| m_j \right|^2 \right]^{1/2} \quad (1)$$

Si la magnitud es una función continua en el tiempo m(t),

$$M_{rms} = \left[\left(\frac{1}{t_2 - t_1} \right) \int_{t_1}^{t_2} \left| m(t) \right|^2 dt \right]^{1/2} \quad (2)$$

La sumatoria o la integral se extienden sobre el intervalo de tiempo para el cuál se desea calcular el valor RMS o, si la función es periódica, sobre cualquier número entero de repeticiones periódicas de la función.

1.1.12. Valor promedio

La media de la magnitud[2]. Si la magnitud está representada por n valores discretos m_j, la magnitud promedio es

[2] Teniendo en cuenta la definición de magnitud descrita anteriormente.

$$\overline{M} = \left(\frac{1}{n}\right)\sum_{j=1}^{j=n} m_j \quad (3)$$

Si la magnitud es una función continua del tiempo m(t),

$$\overline{M} = \left(\frac{1}{t_2 - t_1}\right)\int_{t_1}^{t_2} m(t)dt \quad (4)$$

La sumatoria o la integral se extienden sobre el intervalo de tiempo para el cuál se desea calcular el promedio de la magnitud o, si la función es periódica, sobre cualquier número entero de repeticiones periódicas de la función.

1.2. PROCESADOR DE SEÑALES DIGITALES

Los DSPs han traído nuevas herramientas de análisis al mundo de la electrónica. Con su llegada, no sólo se han incrementado las velocidades de análisis, sino que la precisión y la resolución se han mejorado significativamente. Su distribución a gran escala ha mejorado la eficiencia en el manejo de datos y en el cálculo de resultados. Esto, junto con el desarrollo de rutinas de alta velocidad asegura que la información esté disponible en el menor tiempo posible para aplicaciones que requieren posterior análisis o visualización.

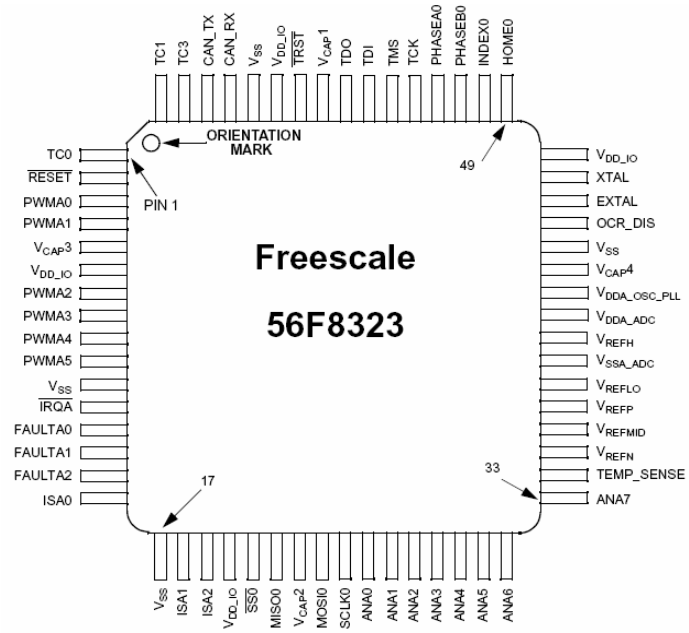
Para la realización de este proyecto se seleccionó el DSP 56F8323 de Motorola de acuerdo a las recomendaciones de los trabajos de grado antecesores basados en DSPs: ***"Elaboración del software para la caracterización de una celda electroquímica utilizando DSP familia 56800 de Motorola"*** por Jean Pierre Amaris y José Alberto López, y, ***"Diseño y construcción de un prototipo de un medidor de armónicos de corriente basado en un procesador de señales digitales (DSP)"*** realizado por Shirley Herrera y Jairo Flórez. El DSP 56F8323, miembro de la familia 56800D de controladores híbridos, dispone de 27 pines de propósito general

y una velocidad de procesamiento de 60MHz. Estos dispositivos incluyen un conjunto de periféricos que son útiles para un gran número de aplicaciones y además combinan la versatilidad de los microcontroladores con la capacidad de procesamiento de los DSPs.

1.2.1. Características del DSP56F8323

El núcleo de la familia 56800E está basado en una arquitectura de tipo *Harvard* que consiste en tres unidades de ejecución operando en paralelo, permitiendo hasta seis operaciones por ciclo de instrucción. El controlador híbrido 56F8323 incluye un conjunto de periféricos estándar programables conformado por dos interfaces de comunicación serial (SCIs), dos Interfaces de periféricos seriales (SPIs), dos Timers, y un FlexCAN. Cualquiera de estas interfaces puede utilizarse como Entradas/Salidas de propósito general (GPIO) si esa función no es requerida.

Figura 1. Diagrama de pines del DSP 56F8323



Hoja de Datos 56F8323 Preliminary Technical Data.

1.2.2. Circuitos periféricos

La versatilidad del DSP 56F8323 está soportada por su conjunto de periféricos. A continuación se enumeran los principales y posteriormente se hace una breve descripción de aquellos utilizados en este proyecto.

- Un Modulador por Ancho de Pulso (PWM) con seis salidas PWM.
- Dos Conversores Análogo / Digital (ADCs) con 12 bits de resolución.
- Dos módulos Timer (TMR) de 16 bits.
- Dos Interfaces de Comunicación Serial (SCIs).
- Dos Interfaces de Periféricos Seriales (SPIs).
- 27 Pines de Propósito General (GPIO).
- Puerto JTAG para programación y depuración en tiempo real.

- **Conversor analógico / digital (ADC)**

El Conversor Analógico / Digital (ADC) consiste de dos ADCs completamente independientes, cada uno con sus circuitos de muestreo y mantenimiento. Las características de este periférico son:

- Resolución de 12 bits.
- Frecuencia máxima del reloj del ADC es 5MHz con un periodo de 200ns.
- Tiempo de conversión mínimo de 1.7µs (588kHz correspondientes a 8,5 ciclos de reloj del ADC).
- Muestreo secuencial o simultáneo.
- Valor de *offset* programable para corregir las muestras.
- Interrupciones opcionales al final de un *scan*, si se excede un límite del rango (superior o inferior) o en los cruces por cero.

- **Timer (TMR)**

Este periférico es utilizado en esta aplicación para la medición del periodo de las señales, dentro de sus principales características se encuentran:

- Cuatro contadores / *timers* de 16 bits.
- Capacidad para contar de forma ascendente y descendente.
- Cada contador tiene capacidad de capturar y comparar.

- **Módulo GPIO**

El módulo GPIO permite acceso directo de lectura y escritura a los valores de los pines y también permite configurarlos para interrupciones externas. En total hay 27 pines divididos en 3 puertos que se pueden configurar como GPIO, en la siguiente tabla se muestra el tamaño de cada puerto así como su función periférica asociada.

Tabla 1. Configuración de los puertos GPIO en el DSP 56F8323

Puerto GPIO	Tamaño del Puerto	Pines disponibles	Función Periférica	Función en Reset
A	12	12	PWM, SPI 1	PWM
B	8	8	SPI 0, DEC 0, TMRA, SCI 1	SPI 0, DEC 0
C	7	7	XTAL, EXTAL, CAN, TMRC, SCI 0	XTAL, EXTAL, CAN, TMRC

Hoja de Datos 56F8323 Preliminary Technical Data, 2004.

2. HARDWARE

El Hardware implementado y utilizado en este proyecto consiste de una tarjeta de desarrollo para el DSP 56F8323 de Motorola de propósito general que se utilizó para realizar las pruebas preliminares al software de adquisición, procesamiento y visualización de las señales. Una pantalla LCD gráfica para la visualización de las formas de onda junto con los resultados de la medición de los parámetros, y, el prototipo final, que consiste de una tarjeta de propósito específico diseñada para integrar la adquisición, acondicionamiento, procesamiento, visualización y programación. Cada uno de estos dispositivos se describe en detalle en las siguientes secciones.

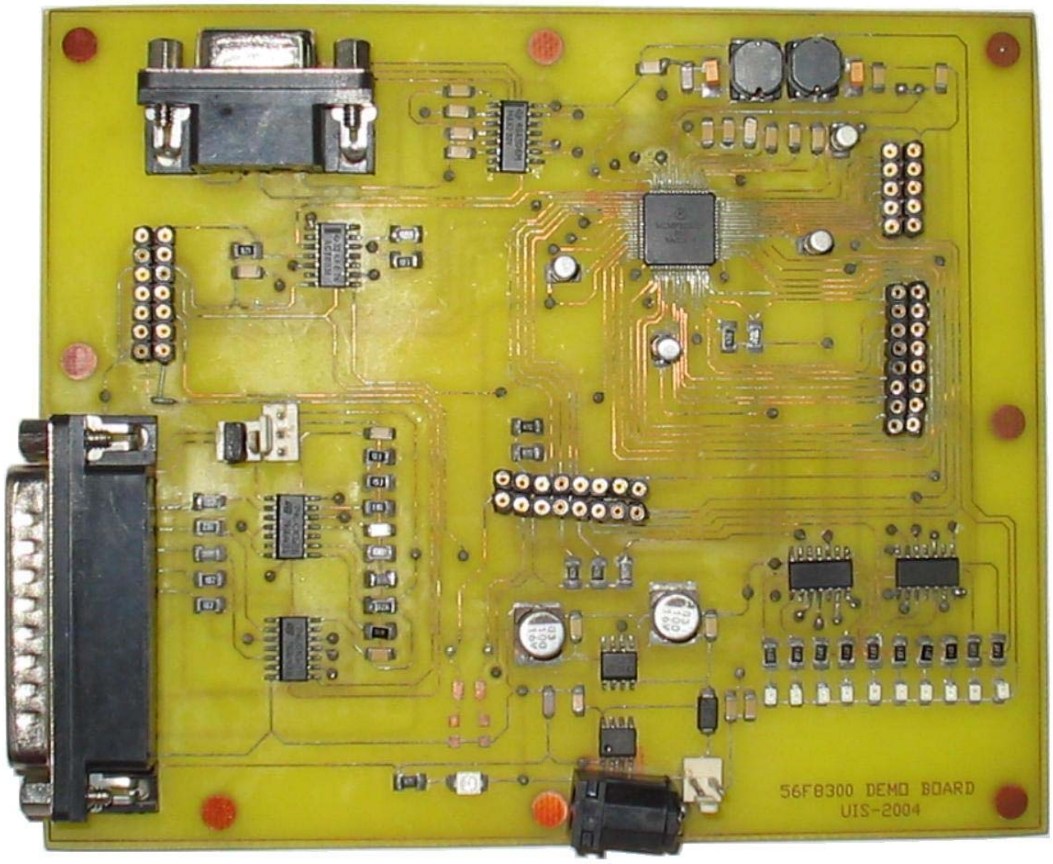
2.1. TARJETA DE DESARROLLO 56F8300

La Tarjeta de Desarrollo 56F8300 le permite al usuario desarrollar sus propias aplicaciones utilizando la herramienta *CodeWarrior*. Esta tarjeta fue implementada de acuerdo a la información proporcionada por el fabricante en los manuales 56F8300 Demonstration Board. MC56F8300DBUM y Demonstration Board. MC56F8300TUM. La información que se presenta en esta sección debe utilizarse conjuntamente con las referencias mencionadas y el diseño esquemático de la tarjeta implementada. A continuación se listan las principales características de la tarjeta de desarrollo:

- CPU 56F8323.
- Puerto GPIO / SERIAL (16 pines).
- Puerto TIMER / PWM (16 Pines).
- Puerto ADC (10 Pines).
- Puerto JTAG / EOnCE (14 Pines).

- Puerto Host JTAG (P1-DB25P).
- Puerto COM1 – SCI a RS-232 (DB9S).
- Conector PWR Jack de alimentación (Voltaje positivo en el centro).
- Indicador de Alimentación.
- 10 Indicadores (LEDs).

Figura 2. Tarjeta de desarrollo 56F8300

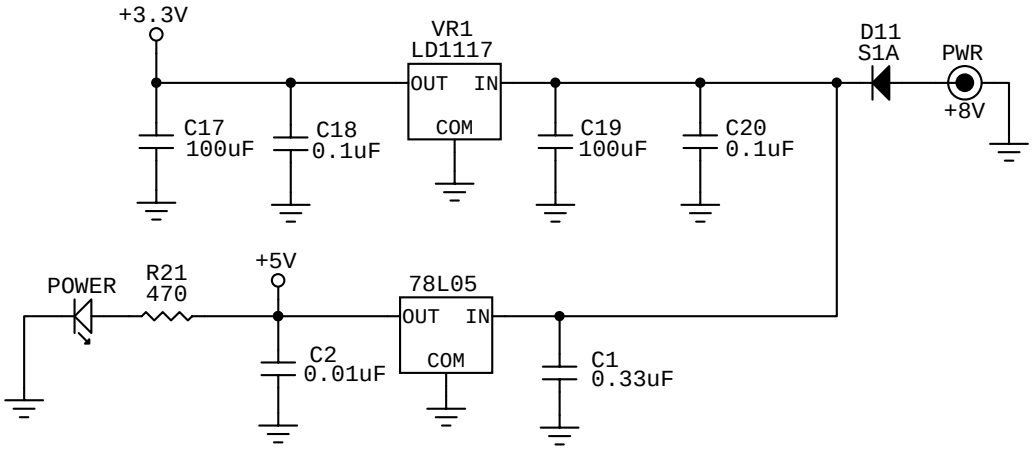


2.1.1. Especificaciones

- 7positivo en el centro).
- Fuentes reguladas en la tarjeta de +5V DC y +3.3v DC.
- Dimensiones: 13.45cms ancho x 11.18cms alto.

En la siguiente figura se muestra el diagrama de alimentación de la tarjeta de desarrollo 56F8300.

Figura 3. Esquema de alimentación de la tarjeta de desarrollo 56F8300



2.1.2. Opciones

El conjunto de opciones de la tarjeta de desarrollo está constituido por un *jumper* para la habilitación de la programación y un *jumper* para seleccionar el nivel de voltaje de interfase del puerto paralelo del PC. Estas opciones se explican en detalle a continuación y en la Figura 12 se puede apreciar su esquema de conexión.

- ***Jumper* HOST_ENABLE**

Este *jumper* habilita la interfase Host JTAG en la tarjeta de desarrollo. Si está instalado (cerrado) se habilitará el modo de depuración del DSP y éste esperará los comandos de programación provenientes del PC. Con el *jumper* desinstalado (abierto) el DSP estará en modo normal y ejecutará el programa contenido en su memoria FLASH.

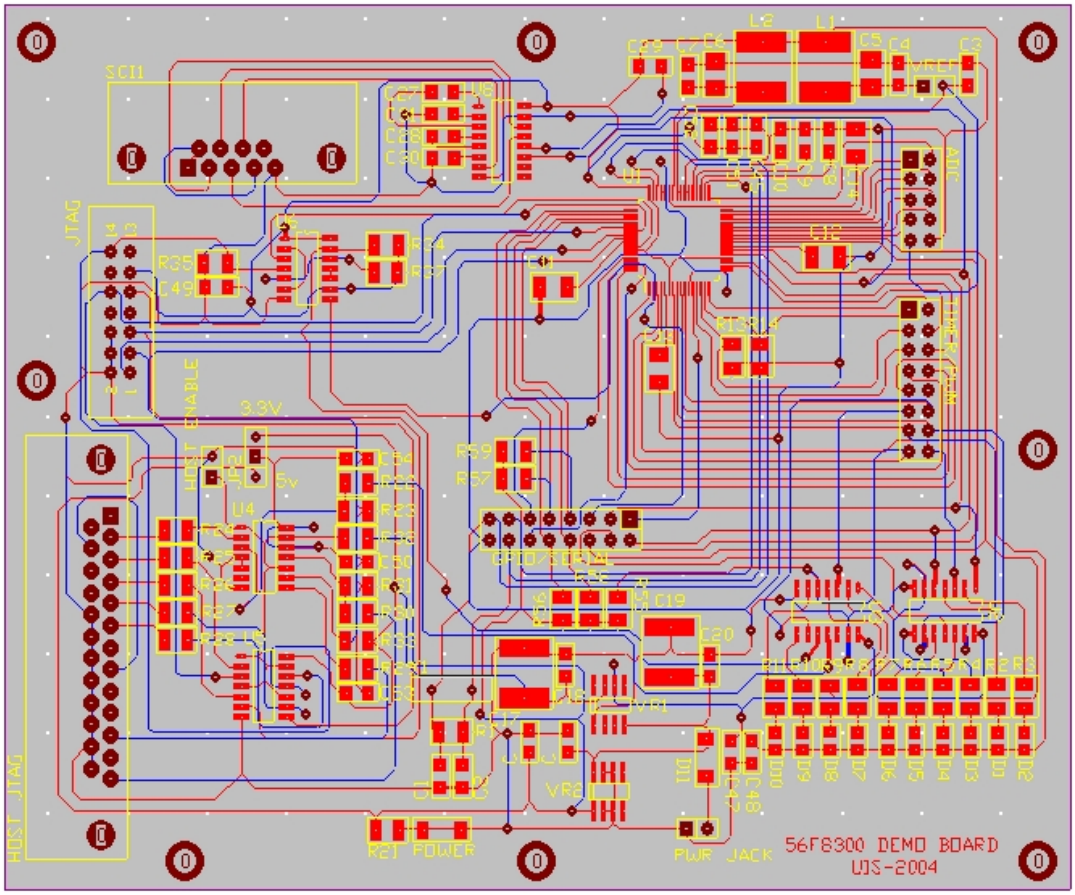
- **JP2 – Nivel de Interfase del Host JTAG**

El *jumper* JP2 selecciona el nivel de interfase del puerto paralelo Host JTAG (P1). Se puede seleccionar entre +3.3V y +5V dependiendo del PC. Para la tarjeta de desarrollo 56F8300 de propósito general se seleccionó +3.3V.

2.1.3. Puertos

La mayoría de los puertos fueron implementados de acuerdo a la información suministrada por el fabricante, sin embargo, se hicieron algunas modificaciones para facilitar el diseño. Estas modificaciones se realizaron en los puertos del ADC y del GPIO / SERIAL y se explican a continuación. La ubicación de los puertos así como su orientación se deben verificar en el diseño esquemático de la tarjeta.

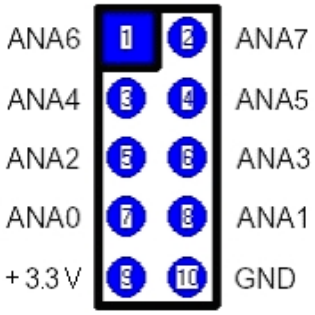
Figura 4. Esquemático Tarjeta de Desarrollo 56F8300



- **Puerto ADC**

Todos los canales del ADC están conectados directamente al DSP, entonces para un óptimo funcionamiento de este periférico y de acuerdo a las especificaciones eléctricas del dispositivo, el voltaje análogo aplicado a estas entradas debe encontrarse en el rango de 0V a +3.3V. En el pin 9 de este puerto se encuentra el voltaje de referencia por defecto que es de +3.3V.

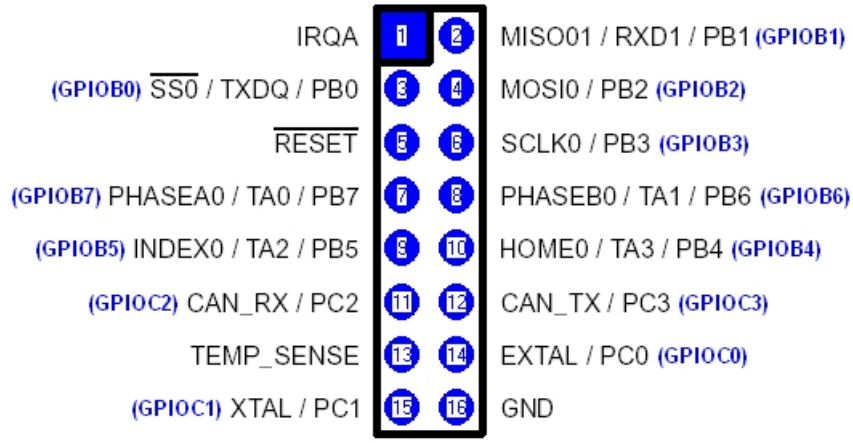
Figura 5. Puerto ADC



- **Puerto GPIO / SERIAL**

En la siguiente figura se aprecia la configuración de los pines del puerto en esta tarjeta. La mayoría de los pines de este puerto pueden configurarse como pines de propósito general, en letras azules se encuentra el nombre del pin GPIO al cuál corresponden. Los pines de este puerto también se encuentran conectados directamente al DSP, por consiguiente el voltaje aplicado a estos pines debe estar dentro del rango de 0V a +6V de acuerdo a las características eléctricas máximas del dispositivo, sin embargo se recomienda operar dentro del rango de 0V a +3.3V.

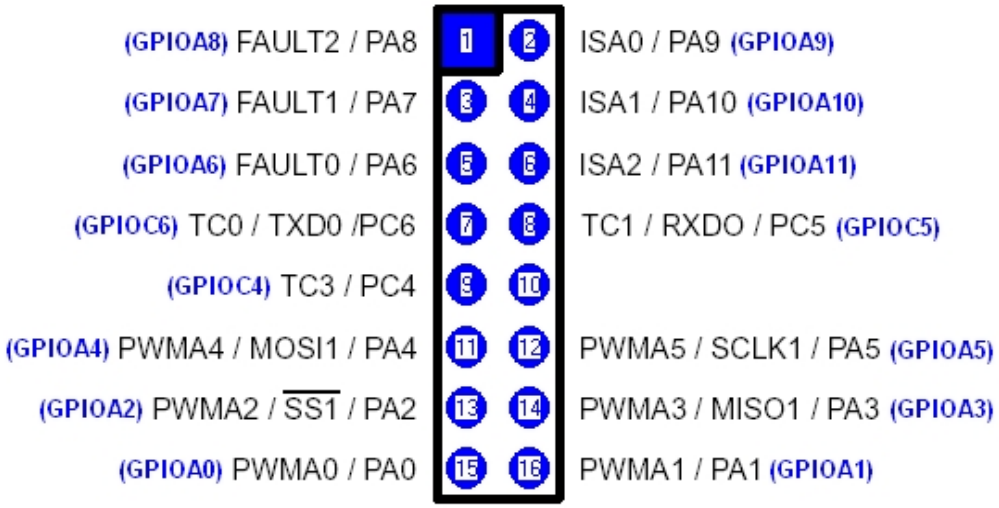
Figura 6. Puerto GPIO / SERIAL



- **Puerto TIMER / PWM**

La mayoría de los pines de este puerto tienen conexiones adicionales dentro de la tarjeta de desarrollo que deben verificarse en el esquemático. Adicionalmente, todos los pines de este puerto se pueden configurar como pines de propósito general, en letras azules se encuentra el nombre del pin GPIO al cuál corresponden. Los pines de este puerto también se encuentran conectados directamente al DSP, por consiguiente el voltaje aplicado a estos pines debe estar dentro del rango de 0V a +6V de acuerdo a las características eléctricas máximas del dispositivo, sin embargo se recomienda operar dentro del rango de 0V a +3.3V.

Figura 7. Puerto TIMER / PWM

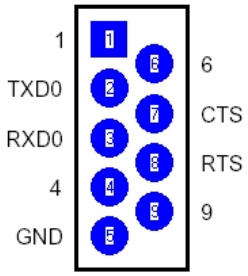


- **Puerto SCI1**

El puerto SCI1 es una interfaz serial que permite conectar el DSP a un periférico externo que utilice el estándar RS-232. Para convertir los niveles TTL a niveles compatibles con el estándar RS-232 se utiliza el circuito integrado MAX232. La conexión básicamente consiste de dos líneas, una de transmisión y una de recepción, a través de las cuáles se lleva a cabo la

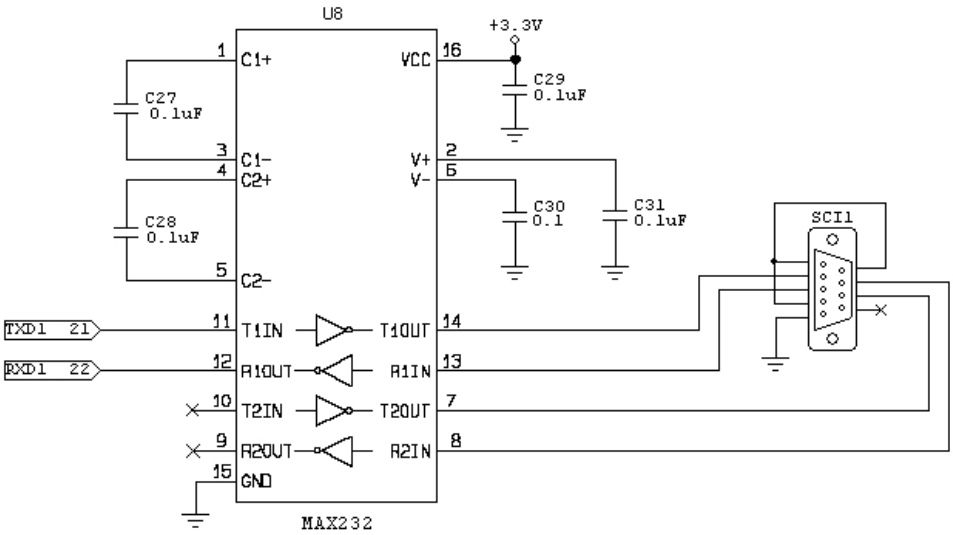
comunicación entre los dos dispositivos. Para conectar la tarjeta de desarrollo al periférico externo, se cuenta con un conector DB9 hembra.

Figura 8. Puerto SCI1



En el siguiente diagrama se puede apreciar el esquema de conexión de la interfase serial. Los pines del DSP utilizados para la transmisión y recepción de datos son SS0 / TXD1 (pin 21) y MISO0 / RXD1 (pin 22).

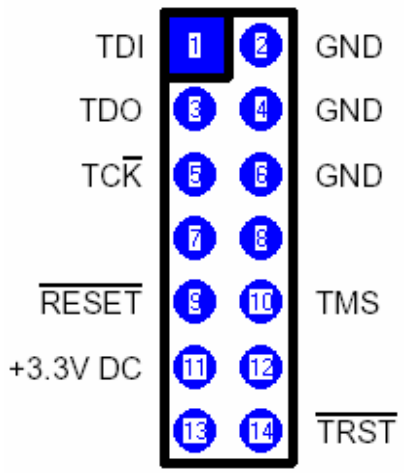
Figura 9. Esquema de conexión de la interfase serial



- **Puerto JTAG / EOnCE**

El puerto JTAG / EOnCE es básicamente un conector de 14 pines que le permite al usuario programar y depurar en tiempo real. Este puerto sólo se encuentra activo si el *jumper* HOST_ENABLE está desinstalado (abierto).

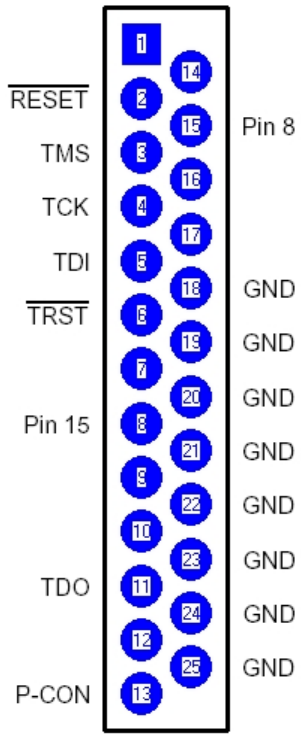
Figura 10. Puerto JTAG / EOnCE



- **Puerto P1 – Host JTAG**

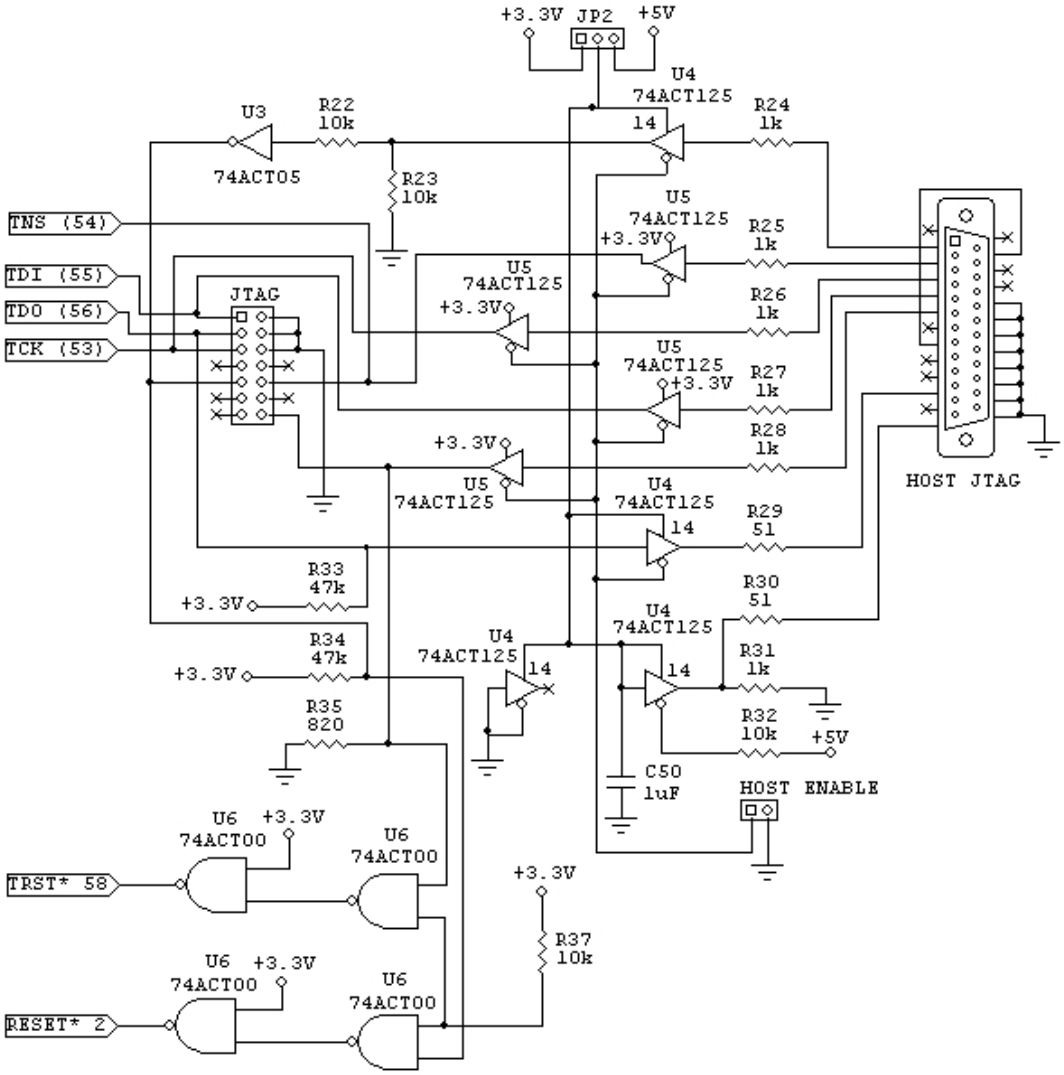
Este puerto permite programar el DSP utilizando la herramienta de desarrollo *CodeWarrior* a través del puerto paralelo del PC. Para habilitar este puerto debe instalarse (cerrarse) el *jumper* HOST_ENABLE en la tarjeta. El conector de la tarjeta es un DB25 macho.

Figura 11. Puerto P1 – Host JTAG



En la siguiente figura se muestra el diagrama de conexión de la interfase de programación. Aquí se pueden apreciar los pines que utiliza el DSP para su programación así como la conexión de los *jumpers* JP2 y HOST ENABLE mencionados anteriormente.

Figura 12. Esquema de conexión de la interfase de programación



2.1.4. Indicadores

La tarjeta de desarrollo cuenta con 10 indicadores (LEDs) de color verde ideales para realizar aplicaciones básicas que le permitan al usuario comprobar el funcionamiento de los pines de propósito general y la utilización de *Embedded Beans*™. Al configurar estos pines para propósito general, se pueden encender/apagar los LEDs aplicando un nivel lógico alto/bajo (1/0) a

cada uno de ellos. En el esquemático de la tarjeta se puede apreciar la ubicación de estos indicadores, y en la siguiente tabla se muestran los pines del DSP a los cuales están conectados.

Figura 13. Esquema de conexión de los Indicadores

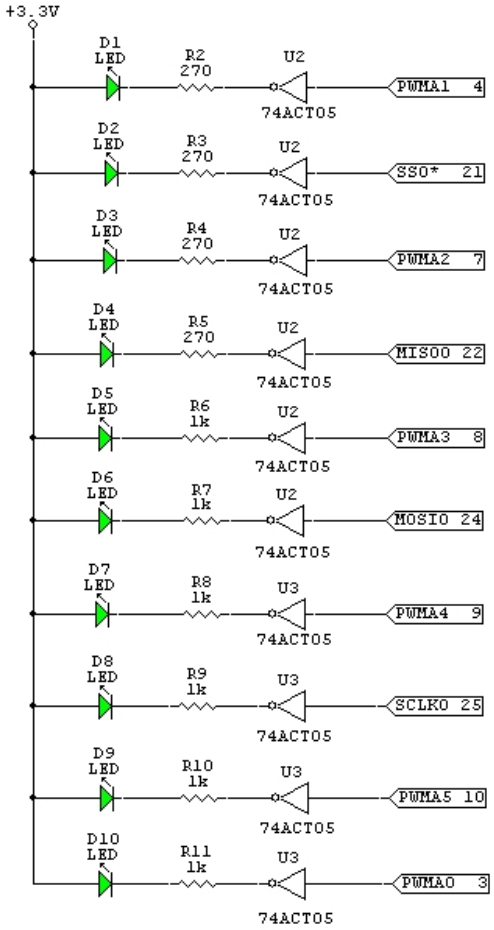


Tabla 2. Indicadores Tarjeta de Desarrollo 56F8300

Indicador	**# Pin**	**Pin 56F8323**
D1	4	PWMA1 / PA1
D2	21	SS0 / PB0

D3	7	PWMA2 / PA2
D4	22	MISO0 / PB1
D5	8	PWMA3 / PA3
D6	24	MOSI0 / PB2
D7	9	PWMA4 / PA4
D8	25	SCLK0 / PB3
D9	10	PWMA5 / PA5
D10	3	PWMA0 / PA0

2.2. PANTALLA GRÁFICA LCD

Para la visualización de la forma de onda de las señales adquiridas junto con los resultados de las mediciones de los parámetros: voltaje pico a pico, frecuencia, valor RMS y valor promedio se utilizó la pantalla gráfica de cristal líquido de referencia HG25504NG-01 de Hyundai de 256 x 128 píxeles de resolución. Se seleccionó este módulo LCD debido a cuatro aspectos importantes: el área de *display*, la capacidad de combinar texto y gráficos, el generador interno de caracteres y su bajo costo[3]. En la siguiente tabla se enumeran las principales características físicas de la pantalla.

Tabla 3. Características de la LCD HG25504NG-01 de Hyundai

Ítem	**Característica**	**Unidad**
Tamaño LCD (W X H X T)	147,0 x 116,0 x 12,0	mm
Área de despliegue(W X H)	127,0 x 70,0	mm
Tamaño de punto (W X H)	0,47 x 0,47	mm

Fuente: Hoja de datos de la pantalla HG25504NG-01 de Hyundai.

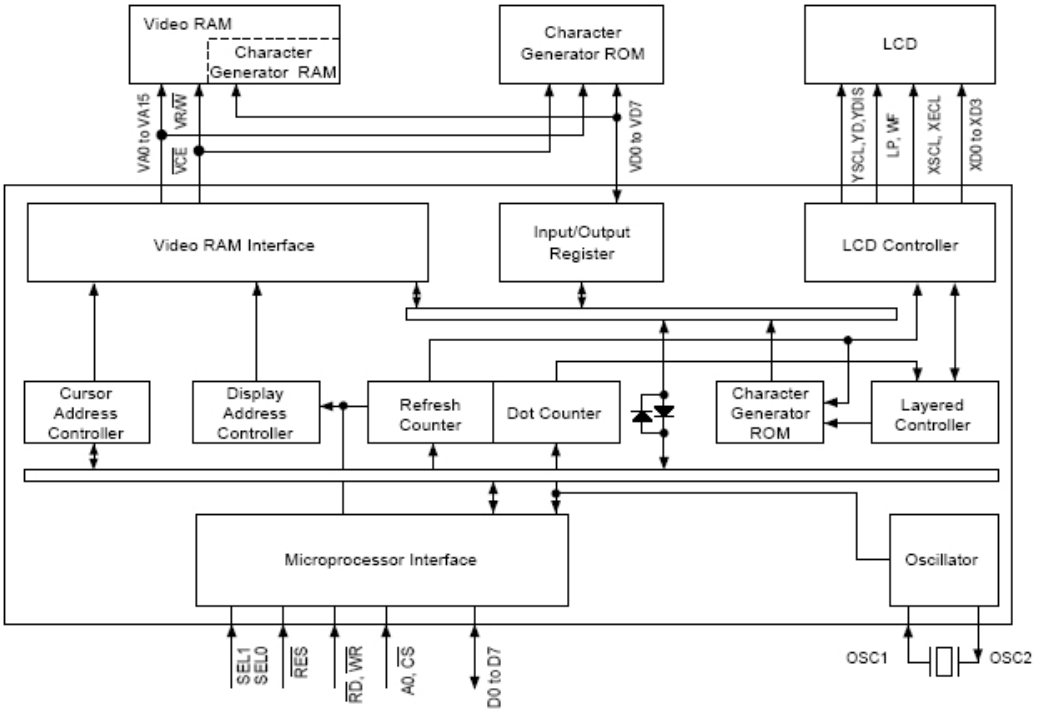
[3] El costo de la LCD es de US$26 sin tener en cuenta los gastos de importación.

Figura 14. LCD Gráfica HG25504NG-01 de Hyundai

Fuente: All Electronics Corp.: www.allelectronics.com.

Esta pantalla posee un controlador gráfico fabricado por S-MOS Systems, de referencia SED1330F cuyo sistema trabaja con base en matrices de 5x7 puntos y puede combinar tanto texto como gráficos gracias a que posee tres capas de pantalla superpuestas que fueron configuradas la primera para texto y las otros dos para gráficos.

Figura 15. Diagrama de bloques de la LCD



Hoja de Datos SED1330F/1335F/1336F's.

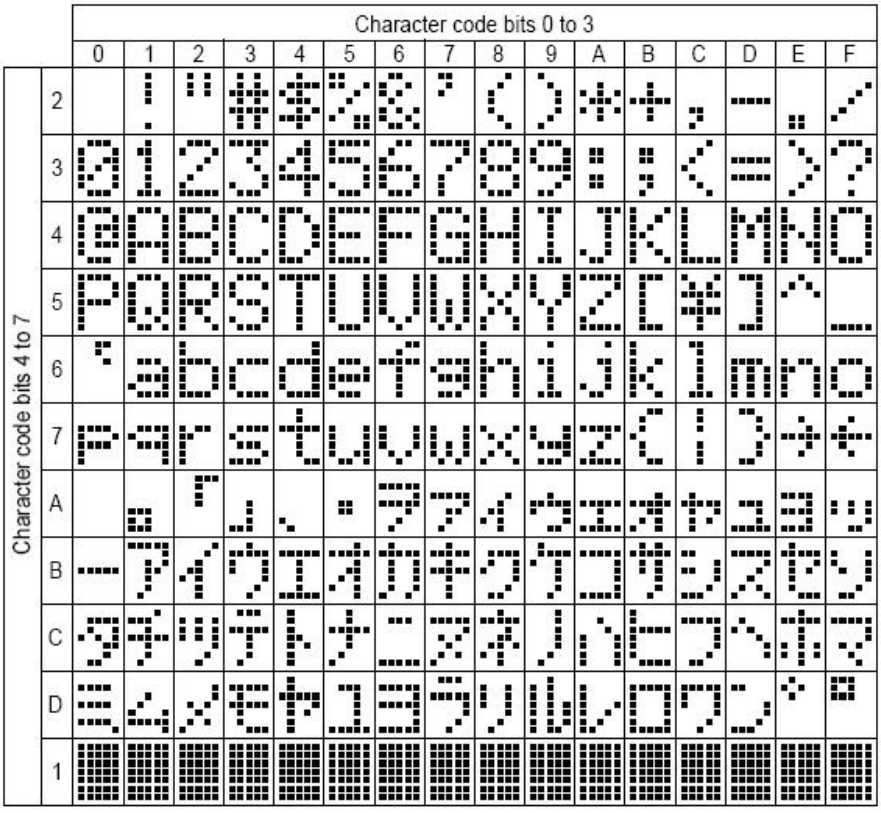
El controlador SED1330F tiene una memoria RAM y ROM con capacidad de 64 y 256 caracteres respectivamente, un bus de datos para la RAM de 8 o 16 bits y posibilidad de interfaz directa con el DSP en paralelo. Si se desea ahorrar pines de propósito general para conectar directamente el DSP al módulo LCD, es posible realizar una interfase serie/paralelo utilizando un registro de desplazamiento.

La pantalla de texto puede desplegar 160 caracteres diferentes, generados a partir del código ASCII interno, en 16 filas por 32 columnas según se muestra en las siguientes figuras:

Figura 16. Distribución de la pantalla de texto

Amaris & López, 2004.

Figura 17. Generador interno de caracteres



Hoja de datos SED1330F/1335F/1336F's.

La pantalla gráfica dispone de 128 líneas por 32 columnas distribuidas de la siguiente manera:

Figura 18. Distribución de la pantalla gráfica

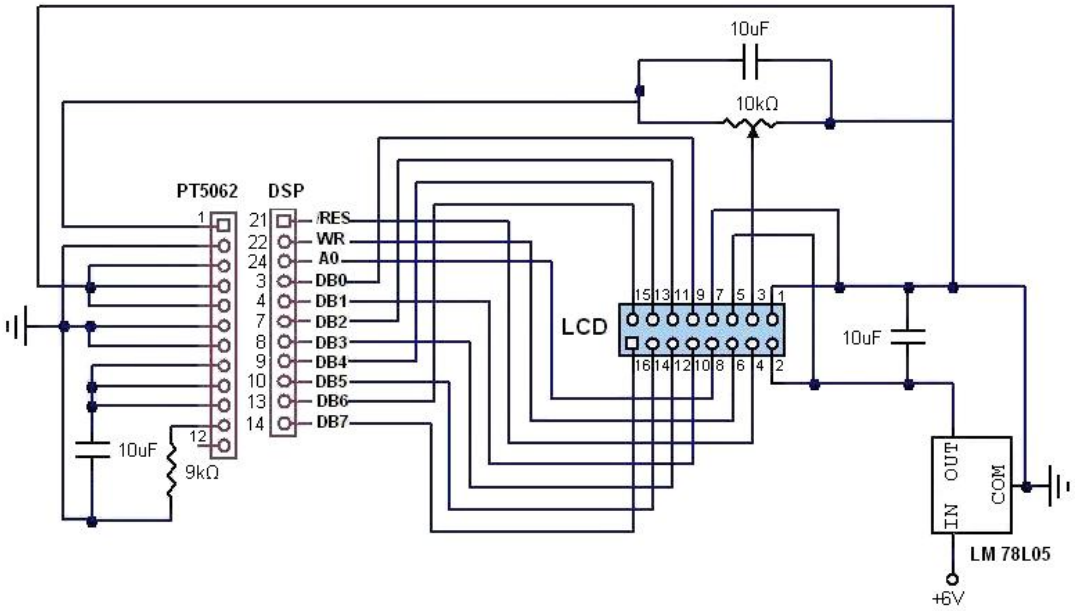
Amaris & López, 2004.

La memoria para texto en la LCD va en la dirección 0x0000 hasta la dirección 0x0FFF (4095 bytes), y la memoria para gráficos comienza en la dirección 0x1000 y termina en la dirección 0x1FFF (4095 bytes).

Según las especificaciones dadas por el fabricante la LCD HG25504NG-01 se alimenta con un voltaje DC entre -10V y -25V, y el controlador SED1330F con un voltaje DC de 5V. En esta aplicación se utilizó el circuito integrado PT5062 fabricado por *Texas Instruments*, que permite obtener una salida dual de ±16V a partir de un voltaje de entrada único de +5V. Este circuito integrado se utiliza para alimentar el circuito de contraste de la LCD (-16V), los amplificadores operacionales y los *switches* análogos utilizados en el prototipo los cuales requieren alimentación dual. Los +5V utilizados para

alimentar este circuito provienen del regulador de voltaje LM78L05, que en su pin de entrada tiene conectada la alimentación de la tarjeta (+8V a 300mA). El contraste de la pantalla esta determinado por un potenciómetro de 10KΩ, que además regula la alimentación de la pantalla de tal forma que se cumpla con la condición de alimentación proporcionada por el fabricante (V_{DD}-V_O= 15.1V donde V_{DD}= 5V). La siguiente figura ilustra las conexiones realizadas para el manejo y alimentación del módulo LCD y en la Tabla 5 se especifican los pines del DSP utilizados para su manejo.

Figura 19. Diagrama de conexión de la LCD.



Como se mencionó anteriormente la pantalla de cristal líquido posee un controlador que se encarga de su funcionamiento. Este utiliza 17 pines para su manejo de los cuales 8 pines son entrada de datos (DB0 a DB7), 5 pines de control y 4 para alimentación. La descripción de estos pines se encuentra en la siguiente tabla.

Tabla 4. Descripción de los pines de la LCD

Nombre del Pin	Nivel	Descripción
FG	0 V	Tierra de carcasa
Vss (GND)	0 V	Tierra
Vdd (Vcc)	+5 V	Fuente de alimentación para la lógica y LCD
Vo	Potenciómetro	Voltaje variable utilizado para el contraste de la LCD
/RES	H/L	Señal de Reset
/RD	H/L	Señal de Lectura
/WR	H/L	Señal de Escritura
/CS	H/L	Señal de elección del chip
A0	H/L	Señal para seleccionar el tipo de datos
DB0	H/L	Dato bit 0
DB1	H/L	Dato bit 1
DB2	H/L	Dato bit 2
DB3	H/L	Dato bit 3
DB4	H/L	Dato bit 4
DB5	H/L	Dato bit 5
DB6	H/L	Dato bit 6
DB7	H/L	Dato bit 7

Hoja de datos de la pantalla HG25504NG-01 de Hyundai.

La interfaz más conveniente para este prototipo es la interfaz paralela por la necesidad de disminuir el “tiempo muerto” del dispositivo. Para el acople de la LCD con el DSP se dispuso de 11 pines que se configuraron como GPIO según la siguiente tabla.

Tabla 5. Configuración de los pines GPIO para el manejo de la LCD

Nombre del Pin en la LCD	No del Pin en el DSP	Nombre del Pin en el DSP
/WR	22	MISO0
A0	24	MOSI0
DB0	3	PWMA0
DB1	4	PWMA1
DB2	7	PWMA2
DB3	8	PWMA3
DB4	9	PWMA4
DB5	10	PWMA5
DB6	13	FAULT0
DB7	14	FAULT1

Para la inicialización de la pantalla, la configuración de las capas, la elección del tamaño del carácter, la configuración del cursor, etc., el controlador dispone de un listado de comandos que se describen a continuación. Adicionalmente en el capítulo 3 se describen los algoritmos implementados para la inicialización y manejo de la pantalla.

Tabla 6. Descripción de comandos del controlador gráfico

<table>
<tr><th>Clase</th><th>Comando</th><th>Valor en Hex</th><th>Descripción</th></tr>
<tr><td>Control del sistema</td><td>SYSTEM SET</td><td>40</td><td>
<table>
<tr><td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table>
Inicializa los dispositivos, activa el tamaño de ventana y de caracteres. Para esto dispone de ocho instrucciones (P1 a P8) por medio de las
</td></tr>
</table>

<table>
<tr><td rowspan="2"></td><td></td><td></td><td>cuales se selecciona la ROM generadora de caracteres internos o externos, se escoge la configuración de la memoria para caracteres, se elige la altura de los mapas de bits de carácter, el método del controlador LCD, el tamaño del carácter horizontal y vertical, el número de bytes por longitud de una línea y el rango de direcciones horizontales de la pantalla virtual.</td></tr>
<tr><td>SLEEP IN</td><td>53</td><td>Suspende cualquier actividad en la LCD.
<table><tr><td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td rowspan="4">Control del Display</td><td>DISP ON/OFF</td><td>58/59</td><td>Enciende o apaga la pantalla.
<table><tr><td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>D=0 Display apagado. D=1 Display encendido.</td></tr></table>Dispone de una instrucción P1 por la cual se habilita o deshabilita el cursor.</td></tr>
<tr><td>SCROLL</td><td>44</td><td><table><tr><td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr></table>Activa el área de despliegue de la pantalla. Posee 10 instrucciones (P1 a P10) mediante las cuales se definen la dirección de inicio del área de despliegue y el número de líneas por bloque de despliegue.</td></tr>
<tr><td>CSRFORM</td><td>5D</td><td><table><tr><td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td></tr></table>Activa la forma y el tamaño del cursor. Tiene dos instrucciones P1 y P2 por medio de la cual se escoge el tamaño horizontal del cursor y el tamaño vertical del cursor.</td></tr>
<tr><td>CGRAM ADR</td><td>5C</td><td>Activa la dirección de comienzo de la RAM del generador de caracteres.
<table><tr><td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td></tr></table></td></tr>
</table>

<table>
<tr><td rowspan="3"></td><td>CSRDIR</td><td>4C a 4F</td><td>Activa la dirección del movimiento del cursor.
<table><tr><td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>CD1</td><td>CD2</td></tr></table><table><tr><td>CD1</td><td>CD2</td><td>Dirección del cursor</td></tr><tr><td>0</td><td>0</td><td>Derecha</td></tr><tr><td>0</td><td>1</td><td>Izquierda</td></tr><tr><td>1</td><td>0</td><td>Arriba</td></tr><tr><td>1</td><td>1</td><td>Abajo</td></tr></table></td></tr>
<tr><td>HDOT SCR</td><td>5A</td><td>Ajusta el tamaño del barrido horizontal de la pantalla por píxeles.
<table><tr><td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr></table>Posee una instrucción P1 para elegir el número de píxeles para el área de despliegue.</td></tr>
<tr><td>OVLAY</td><td>5B</td><td>Selecciona la composición en capas de la pantalla y el modo de la pantalla texto/gráfica por medio de una instrucción adicional P1.
<table><tr><td>DB7</td><td>DB6</td><td>DB5</td><td>DB4</td><td>DB3</td><td>DB2</td><td>DB1</td><td>DB0</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td></tr></table></td></tr>
<tr><td rowspan="2">Drawing Control</td><td>CSRW</td><td>46</td><td>Escoge la posición del cursor.</td></tr>
<tr><td>CSRR</td><td>47</td><td>Lee desde la dirección del registro del cursor.</td></tr>
<tr><td rowspan="2">Memory Control</td><td>MWRITE</td><td>42</td><td>Escribe en la memoria de la pantalla.</td></tr>
<tr><td>MREAD</td><td>43</td><td>Lee desde la memoria de la pantalla.</td></tr>
</table>

Hoja de datos SED1330F/1335F/1336F's.

De acuerdo a las instrucciones mencionadas anteriormente la pantalla se configuró con las siguientes características:

- ROM Generadora interna habilitada.
- Tamaño del carácter horizontal de 8 píxeles.
- Tamaño del carácter vertical de 8 píxeles.

- 128 líneas por trama.
- La pantalla de texto inicia en la dirección 0x0000.
- La pantalla de gráfica inicia en la dirección 0x1000.
- Pantalla de texto y pantalla gráfica activada.
- Cursor ubicado inicialmente en la primera fila de la primera columna.
- Movimiento del cursor hacia la derecha.

2.3. PROTOTIPO FINAL

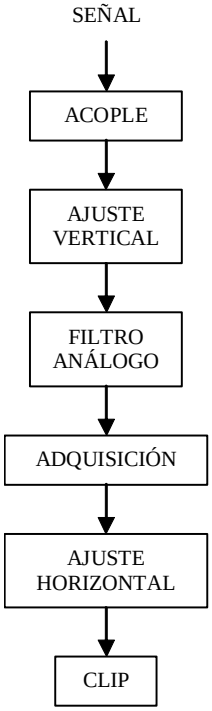
El prototipo final fue implementado para integrar en una sola tarjeta la adquisición, el acondicionamiento, el procesamiento y la visualización de las formas de onda, así como la programación del DSP.

Figura 20. Prototipo final *DIGISCOPE*.

La siguiente figura muestra un diagrama de bloques del canal del osciloscopio. Casi todas las funciones del canal son bastante conocidas y se explican con detalle en secciones posteriores, sin embargo, antes de

continuar es importante definir la función CLIP. Esta función excluye puntos de la señal que están por fuera del área de *display* de la pantalla gráfica.

Figura 21. Diagrama de bloques canal del osciloscopio.



Para el manejo por parte del usuario del prototipo final éste cuenta con tres botones cuyas funciones se distribuyen de la siguiente manera:

Tabla 7. Funciones de los botones del prototipo final.

Nombre del Botón	Función
ON/OFF	Energiza/ Desenergiza el Prototipo *DIGISCOPE*
Reset DSP	Reinicia el DSP
Reset LCD	Reinicia la Pantalla Gráfica

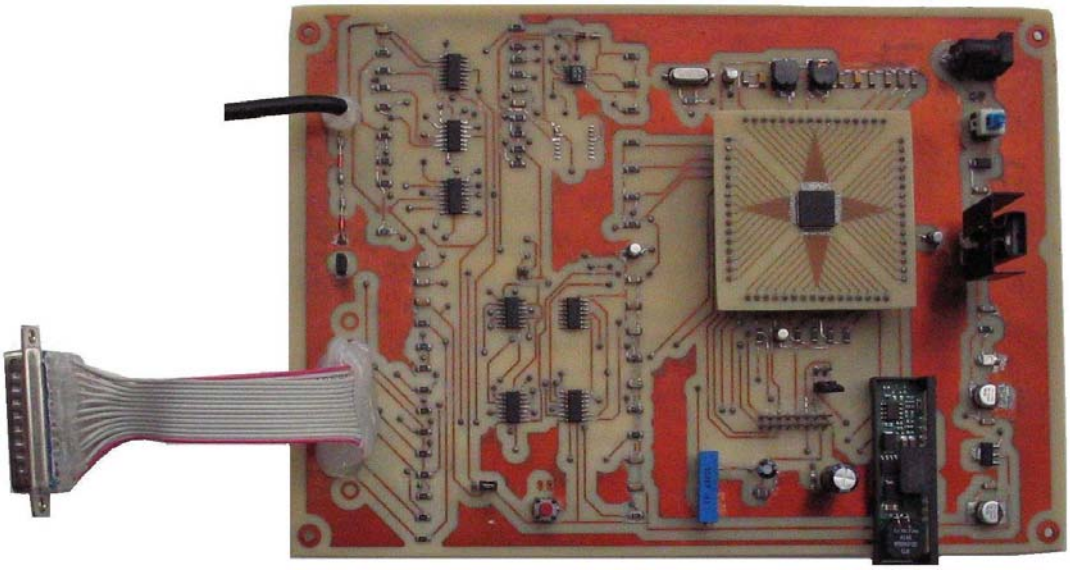
El botón *Reset* LCD fue colocado debido a que en algunos casos la pantalla gráfica no se inicializa correctamente, tal como se muestra a continuación:

Figura 22. Inicialización incorrecta de la pantalla LCD gráfica.

2.3.1. ESPECIFICACIONES

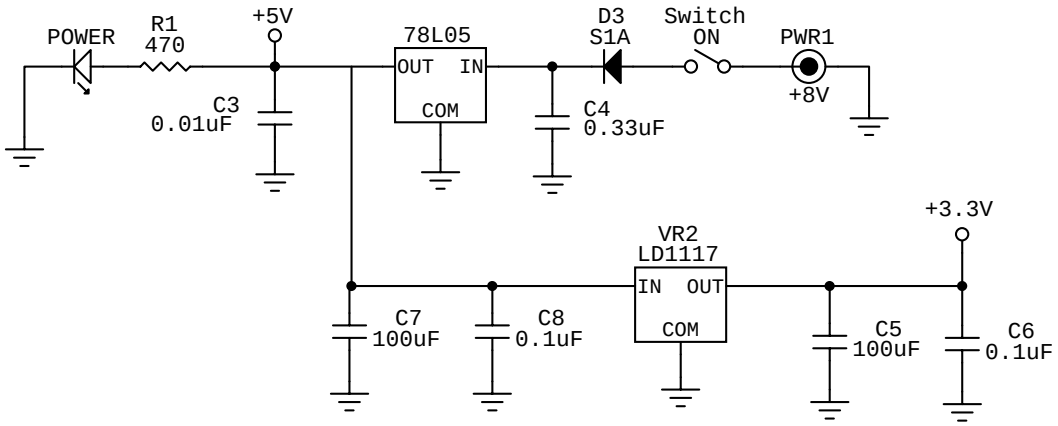
- Voltaje de Alimentación: +8V, 300mA.
- Número de Canales: 1.
- Voltaje máximo de la señal de entrada: ±15V.
- Frecuencia máxima de la señal de entrada: 20KHz.
- Función *AUTOSET* (Ajuste Automático).

Figura 23. Tarjeta de Propósito Específico *DIGISCOPE*.



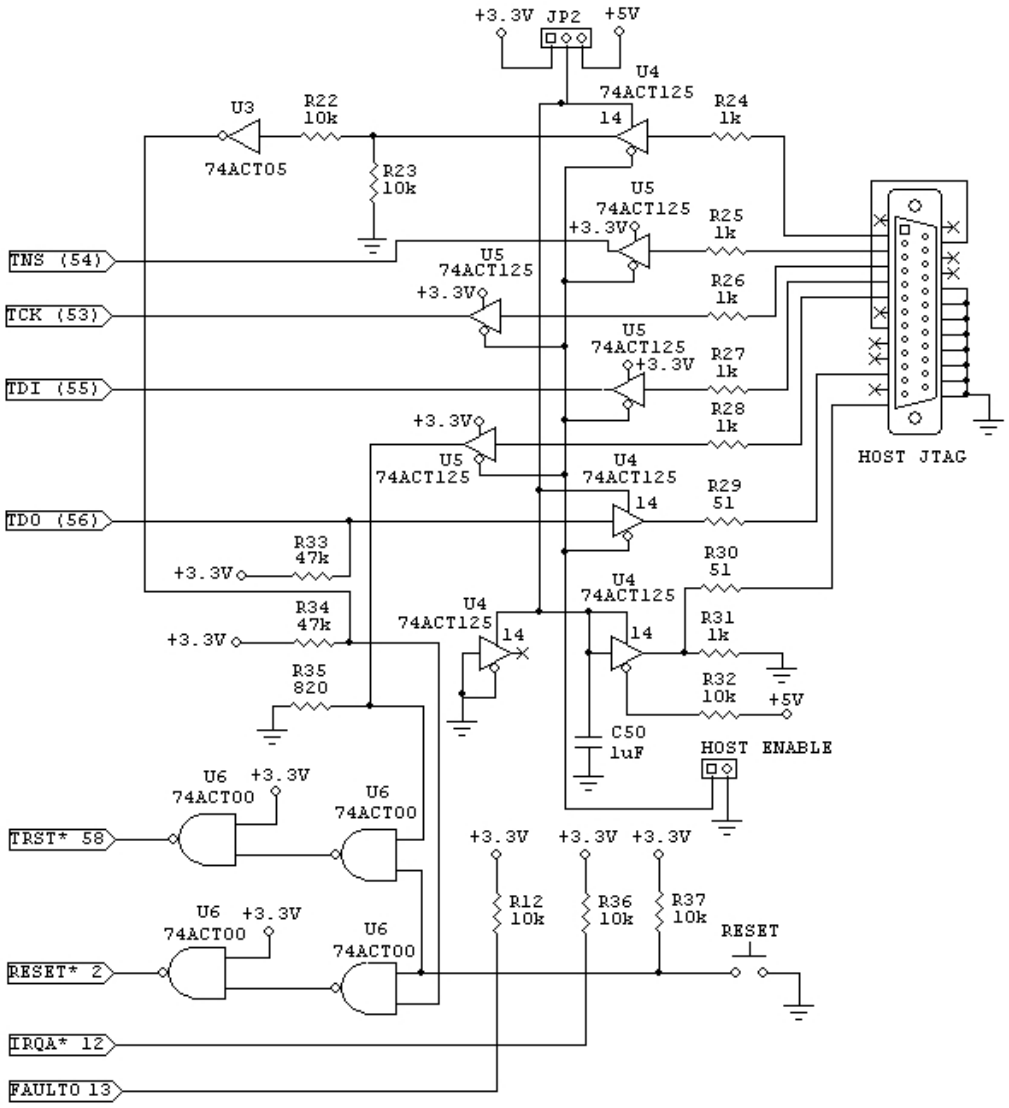
En la siguiente figura se ilustra el esquema de alimentación de la tarjeta del prototipo final *DIGISCOPE*

Figura 24. Esquema de alimentación del prototipo *DIGISCOPE*



La interfase de programación fue modificada con respecto a la implementada en la tarjeta de desarrollo. En esta ocasión se incluyó el pulsador RESET y los pines IRQA* y FAULT0 del DSP se conectaron a 3,3V a través de dos resistencias de 10KΩ. También se eliminó el puerto JTAG descrito en la sección 2.1.3. En la siguiente figura se puede apreciar el esquema de conexión de esta etapa para el prototipo final.

Figura 25. Interfase de programación en el prototipo final



2.3.2. ACONDICIONAMIENTO DE SEÑALES

El proceso de acondicionamiento de las señales consiste de 4 etapas: acople, atenuación, ajuste, y filtrado.

- **Acople, atenuación y ajuste**

La etapa de acople permite eliminar la componente DC de la señal, en caso que la tenga, haciendo circular la señal por un condensador de paso. La selección del tipo de acople se hace de forma automática mediante un *switch* (interruptor) analógico controlado por el DSP. En el capítulo 3 se explica en detalle este procedimiento.

Las etapas de atenuación y ajuste permiten acondicionar las señales de entrada para que éstas no sobrepasen los límites de voltaje permitidos por el ADC del DSP. El rango de voltaje permitido por el ADC se encuentra entre 0V y +3.3V. En esta aplicación las señales de entrada se limitaron a ±15V debido a la restricción en los voltajes de alimentación de los dispositivos utilizados en la etapa de atenuación (*op-amps* y *switches* analógico), lo que significa que éstas se deben atenuar y posteriormente ajustar sumándoles una componente de DC para que no tomen valores negativos.

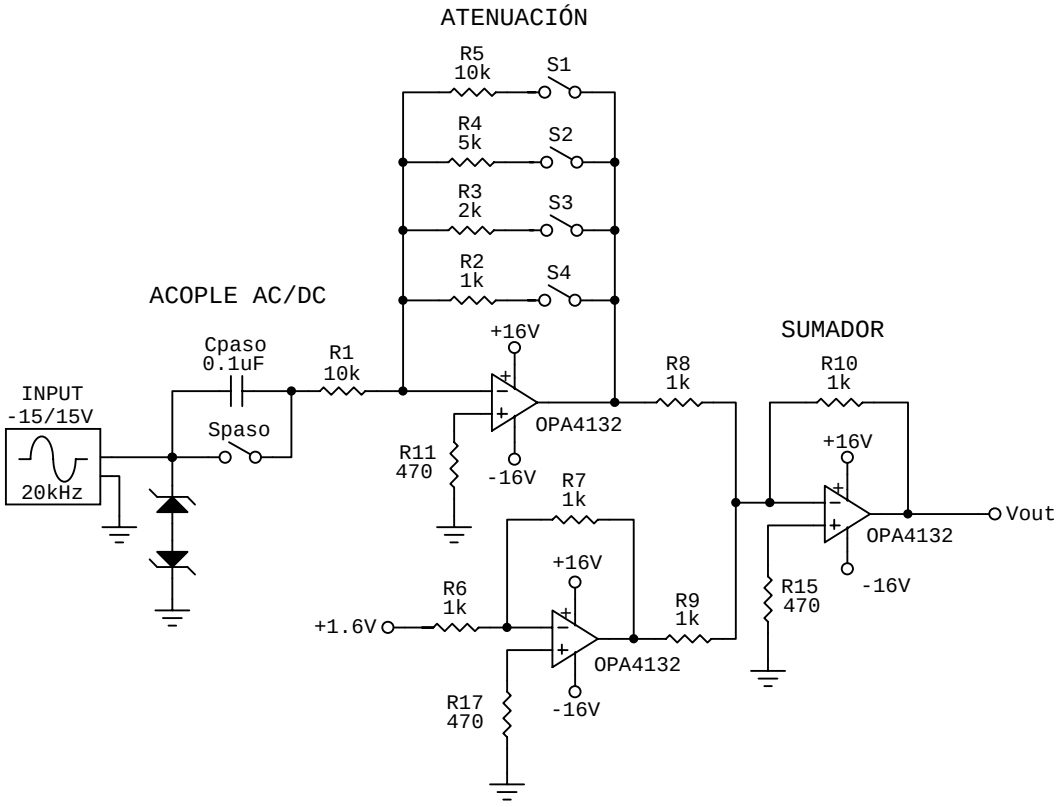
Debido a que se implementó un sistema de *AUTOSET* (ajuste automático), la atenuación de la señales no puede ser fija, ya que esto prácticamente eliminaría las señales de poca amplitud si se toma como referencia para elegir el nivel de atenuación las señales de mayor amplitud (±15V). Para realizar esta función de *AUTOSET* se utilizaron unos *switches* análogos (DG441 Fabricados por MAXIM) controlados por el DSP con los que se selecciona el nivel de atenuación dependiendo de los picos máximo y mínimo de la señal. Estos niveles de atenuación son de 1x para señales de amplitudes menores a 1.5V, de 2x para señales de amplitudes comprendidas

entre 1.5V y 3V, de 5x para aquellas cuya amplitud se encuentran entre 3V y 7.5V, y, de 10x para señales de amplitudes entre 7.5V y 15V. Se seleccionó este circuito integrado debido a su rápido tiempo de respuesta (250nS para t_{ON} y 70nS para t_{OFF}) y a su rango de voltaje de la señal de entrada limitado sólo por el voltaje de alimentación (±16V). Cada nivel de atenuación garantiza que la señal de voltaje no sobrepase el rango de ±1.5V a su salida.

Posteriormente se debe sumar la componente de DC de +1.6V, asegurando así que la señal que entre al conversor quede ubicada dentro de los límites permitidos de voltaje, en este caso entre 0.1V y +3.1V. La señal de 1.6V DC proviene del circuito integrado TPS72216, que es un regulador de voltaje de 1.6V fabricado por *Texas Instruments* seleccionado debido estabilidad. Los amplificadores operacionales corresponden a la referencia OPA4132, también fabricados por *Texas Instruments*. Estos amplificadores poseen alta impedancia de entrada lo que garantiza una corriente de entrada máxima de 50pA, amplio ancho de banda (8MHz), bajo ruido (8nV/$\sqrt{Hz}$), baja distorsión (0.00008%) y amplio rango de alimentación (±2.5V a ±18V). En esta aplicación el integrado OPA4132 se alimentó con ±16V.

Como protección adicional se utilizó una configuración de diodos zener (15V, ½ W) a la entrada del canal para limitar el voltaje de entrada a ±15V. La siguiente figura muestra el circuito de acople, atenuación y ajuste.

Figura 26. Circuito de atenuación y ajuste



Los *switches* análogos son controlados por el DSP según la siguiente tabla.

Tabla 8. Pines GPIO utilizados para controlar los *switches* análogos

GPIO	Pin DSP	Switch
GPIOB3	25	Switch Spaso: C_{paso}
GPIOB4	49	Switch S1: Atenuación 1x
GPIOB5	50	Switch S2: Atenuación 2x
GPIOB6	51	Switch S3: Atenuación 5x
GPIOB7	52	Switch S4: Atenuación 10x

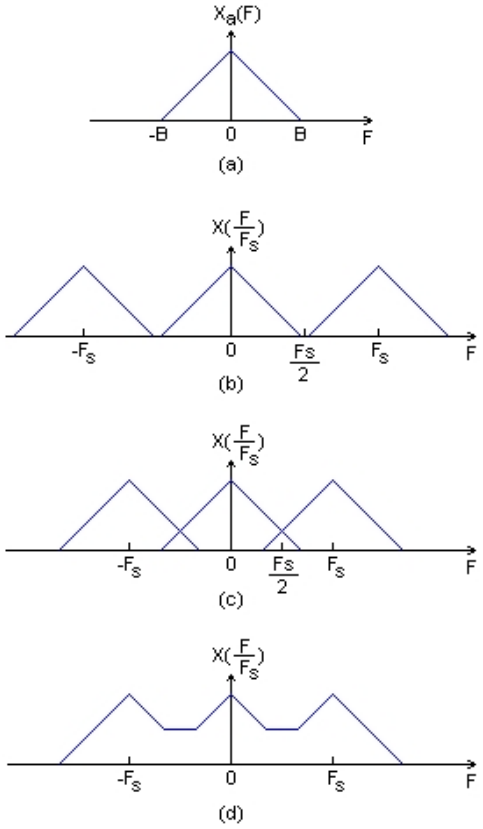
- **Filtro *Anti-Aliasing***

El *aliasing* o solapamiento es un problema que se presenta cada vez que una señal análoga se muestrea para convertirla en una señal digital. Esto ocurre cuando la frecuencia de muestreo no es lo suficientemente alta con respecto a la componente de frecuencia más alta de la señal análoga. Cuando la frecuencia no es lo suficientemente alta entonces se pierde información espectral de la señal.

Para explicar el efecto es conveniente hacer un análisis en el dominio de la frecuencia. De acuerdo al teorema del muestreo, si se tiene una señal de banda limitada cuya mayor frecuencia es B Hz, esta podrá recuperarse a partir de sus muestras si se cumple que la Frecuencia de Muestreo, $F_s \geq 2B$.

Si se tiene una señal análoga de banda limitada $x_a(t)$ cuyo espectro, dado por la transformada de *Fourier* $X_a(F)$, es cero para $|F| \geq B$ como se muestra en la figura (a) y se escoge una frecuencia de muestreo F_s, tal que $F_s \geq 2B$ entonces no se producirá el efecto de solapamiento como se aprecia en la figura (b). Sin embargo, si se escoge una frecuencia de muestreo F_s, tal que $F_s < 2B$, la repetición periódica de $X_a(F)$ producirá un solapamiento espectral como se ilustra en las figuras (c) y (d). Esto significa que algunas componentes de frecuencia se superpondrán a otras, impidiendo la recuperación fiel de la señal a partir de sus muestras.

Figura 27. Análisis espectral de la señal analógica $x_a(t)$

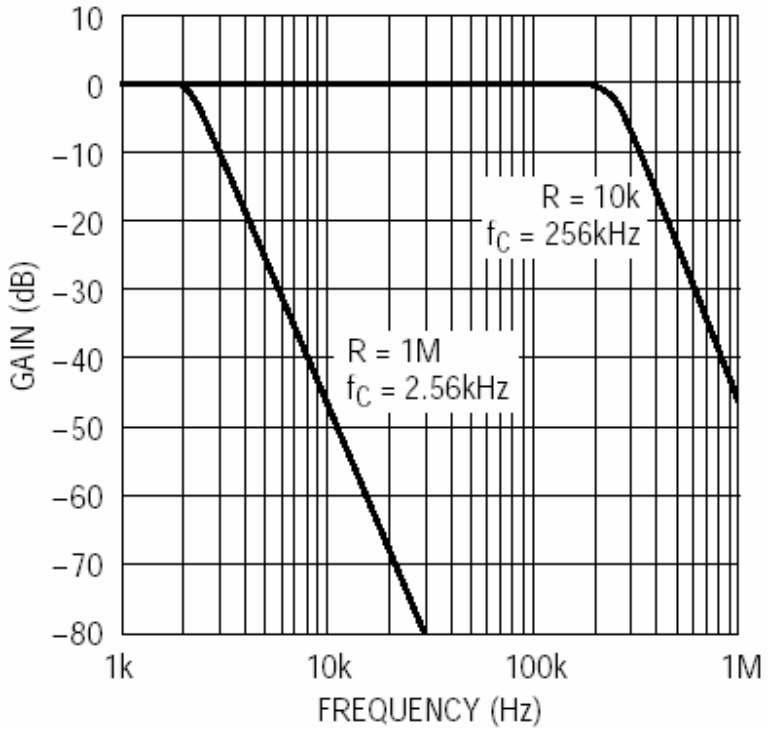


PROAKIS & MANOLAKIS, 1998.

En esta aplicación la frecuencia de muestreo F_s es de 420KHz y la frecuencia más alta de las señales análogas se limitó a 20KHz para poder obtener suficientes muestras que permitan una visualización óptima de la señal en la pantalla y una medición precisa de sus parámetros. Esta limitación garantiza que se cumpla el teorema del muestreo, ya que F_s>2B. Sin embargo, para asegurar que la frecuencia máxima de la señal no sobrepase los 20KHz se utilizó un filtro *anti-aliasing* con frecuencia de corte igual a 25.6KHz. La frecuencia de corte es un poco superior para evitar atenuar las componentes de frecuencia de 20KHz.

Para el diseño de este filtro se utilizó el circuito integrado LTC1563-2 que es un filtro *Butterworth* de cuarto orden. Este tipo de filtro tiene una respuesta máximamente plana en la banda de paso, una característica importante en esta aplicación, y una caída de -80dB por década en la banda de transición, lo que garantiza la eliminación del efecto de solapamiento. Con el propósito de ilustrar la característica de la respuesta en frecuencia del circuito integrado utilizado se incluyó la siguiente figura, que se encuentra en la hoja de datos del dispositivo. Esta figura muestra la respuesta en frecuencia para frecuencias de corte de 2,56KHz y 256KHZ.

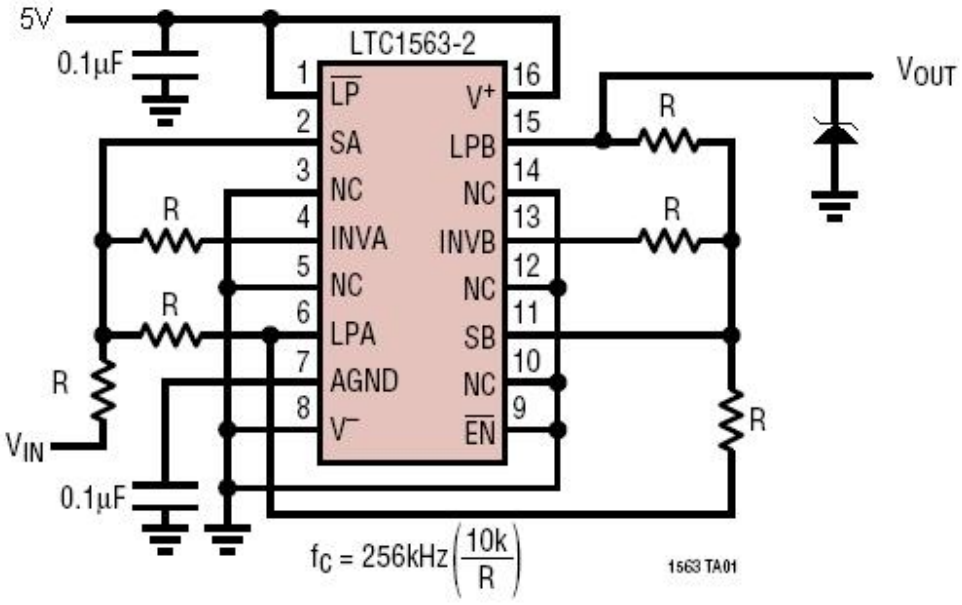
Figura 28. Respuesta en frecuencia del circuito integrado LTC1563-2



Hoja de Datos LTC1563-2.

La implementación de este filtro utilizando el circuito integrado mencionado es bastante sencilla, ya que la frecuencia de corte se puede ajustar con un arreglo de resistencias y capacitores como se muestra en la Figura 29. En esta configuración también se puede apreciar un diodo zener de 3.3V a la salida del filtro, utilizado para asegurar que el voltaje aplicado al conversor análogo / digital de DSP no sobrepase este voltaje. Este circuito integrado tiene como características: bajo ruido (36μV_{RMS} para f_c=25.6KHz), precisión en la frecuencia de corte (<±2%) y *offset* <1mV.

Figura 29. Circuito utilizado para la implementación del filtro



Hoja de Datos LTC1563-2.

De acuerdo a la información del fabricante (Linear Technology), la frecuencia de corte se ajusta variando el valor de la resistencia R, según la siguiente ecuación.

$$R = 10K\left(\frac{256KHz}{f_c}\right) \quad (5)$$

El valor de la resistencia (R) seleccionado fue de 100KΩ, lo que permite obtener una frecuencia de corte de 25,6KHz.

Posteriormente, mediante software se implementó un filtro digital más riguroso para mejorar la respuesta en frecuencia del sistema y disminuir el ancho de la banda de transición.

3. SOFTWARE

Los algoritmos diseñados para la implementación de este proyecto fueron desarrollados utilizando la plataforma *CodeWarrior™ Development Studio for Motorola™ DSP56800/E Hybrid Controllers, versión 6.1.* y el *Processor Expert™*. El *Processor Expert™* está diseñado para el rápido desarrollo de aplicaciones que se implementan utilizando componentes llamados *Embedded Beans™*. Los *Beans* encapsulan los elementos básicos del sistema como la CPU, los periféricos del chip, y algoritmos de software puros y permiten su manipulación a través de propiedades, métodos y eventos.

3.1. CONFIGURACIÓN DEL DSP

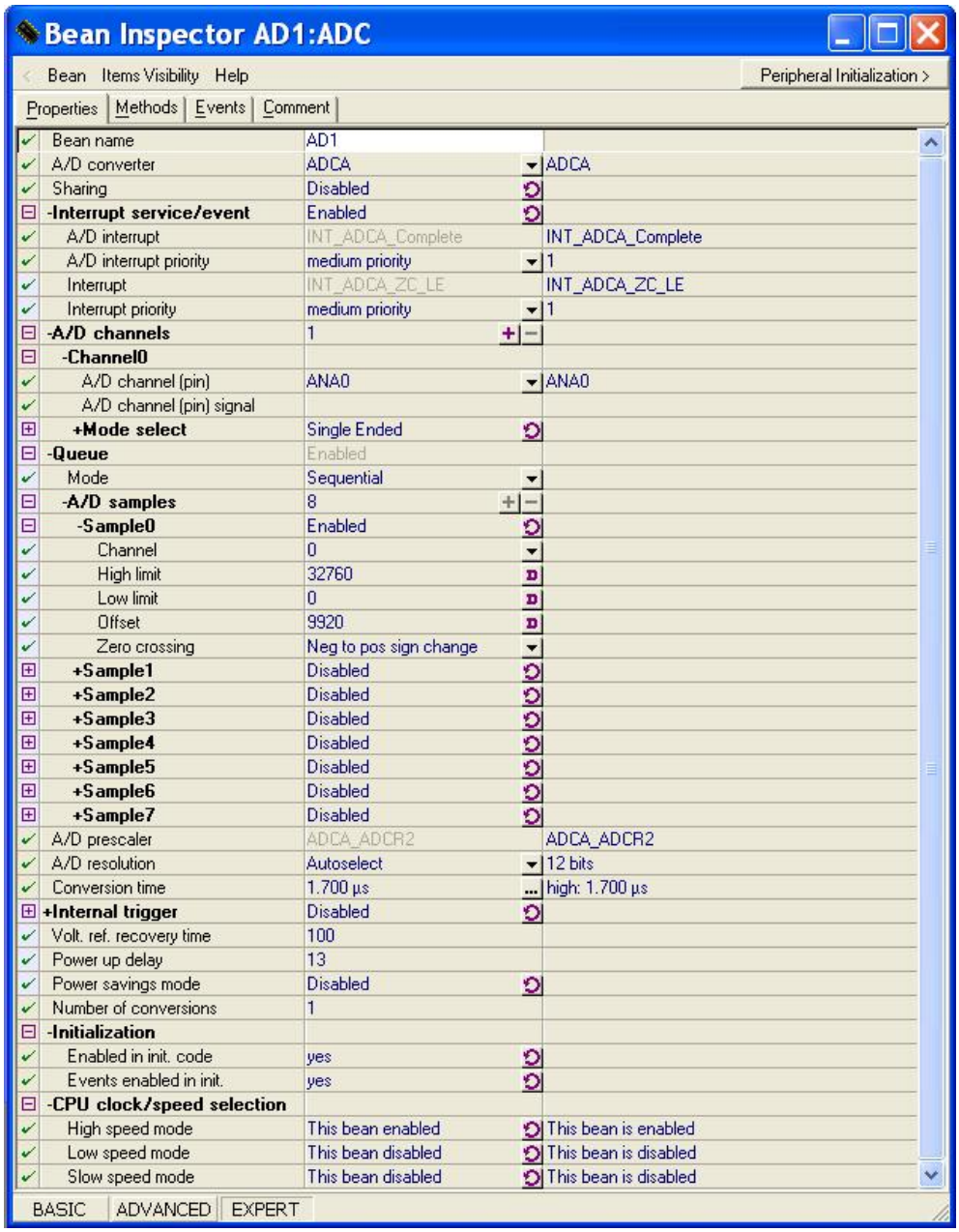
Para la configuración del DSP se utilizaron *Embedded Beans™* así como instrucciones adicionales para manipular los registros de los periféricos y reconfigurarlos según las necesidades de la aplicación.

3.1.1. Conversor Analógico / Digital (ADC)

El Conversor Análogo / Digital se configuró inicialmente con el Bean ADC que se encuentra bajo la categoría *CPU Internal Peripherals/Converter/ADC*. Como se muestra en la siguiente figura, este *Bean* permite configurar todas las opciones del ADC del DSP, así como la utilización de Métodos (funciones) y la habilitación de eventos. El método que se utilizó fue *Measure*, que como indica su nombre realiza una medición (conversión) de todos los canales seleccionados en las opciones, y se habilitó el evento *OnZeroCrossing*, que permite detectar los cruces por un nivel de referencia. Es importante mencionar que se debe ajustar el valor del *offset* para que el

conversor reste este valor del resultado de la conversión antes de guardarlo en los registros de resultados (ADRSLT0–7) y efectivamente pueda detectar los cruces por el nivel de referencia.

Figura 30. Propiedades del *Bean* ADC



Como se había mencionado antes, fue necesario utilizar los registros del ADC para reconfigurarlo de acuerdo a las varias etapas del proceso de adquisición de la señal. A continuación se exponen los registros utilizados.

- **Registro *ADC Control Register 1* (ADCR1)**

Este registro permite inicializar un *scan*, detenerlo, habilitar las interrupciones debidas al fin de un *scan*, cruces por cero, cruces por límites superior e inferior, configuración de los canales y configuración del modo de *scan*.

Figura 31. Registro *ADC Control Register 1* (ADCR1)

Base + $0	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	STOP	0	SYNC	EOSIE	ZCIE	LLMTIE	HLMTIE	CHNCRG				0	SMODE		
Write			START													
Reset	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	1

MC56F8300 Peripheral User Manual, 2004.

- **Registro *ADC Control Register 2* (ADCR2)**

Este registro permite seleccionar el divisor del reloj del ADC, ya que éste se debe manejar a una rata más lenta que la del reloj del sistema, esto quiere decir que se debe seleccionar un valor del divisor para que el ADC trabaje dentro de los límites especificados. En esta aplicación la frecuencia de reloj del IPBus es de 60MHz, y la frecuencia de reloj del ADC debe ser máximo de 5MHz, entonces se requiere un valor mínimo de 5 en el campo DIV de este registro.

Figura 32. Registro *ADC Control Register 2* (ADCR2)

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	0	0	0	DIV				
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

MC56F8300 Peripheral User Manual, 2004.

La siguiente ecuación permite calcular la frecuencia a la cuál trabajará el ADC.

$$F_{ADC} = {(F_{IPR})}/{2N} \quad (6)$$

Donde N es el Valor del Divisor del Reloj = N = DIV + 1. F_{ADC} es la Frecuencia del Conversor Analógico / Digital y F_{IPR} es la Frecuencia de Reloj del Bus de Interfase de Periféricos.

- **Registro *ADC Zero Crossing Control Register* (ADZCC)**

Este registro permite monitorear los canales seleccionados y determinar la dirección de los cruces por cero. Cada campo de dos bits permite seleccionar, en el respectivo canal, el tipo de monitoreo que se realizará.

Figura 33. *Registro ADC Zero Crossing Control Register* (ADZCC)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	ZCE7		ZCE6		ZCE5		ZCE4		ZCE3		ZCE2		ZCE1		ZCE0	
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MC56F8300 Peripheral User Manual, 2004.

Las opciones de este registro se presentan a continuación.

- 00 = Cruces por Cero deshabilitados.
- 01 = Habilitados para cambios de signo de positivo a negativo.
- 10 = Habilitados para cambios de signo de negativo a positivo.
- 11 = Cruce por cero habilitado para cualquier cambio de signo.

- **Registro *ADC Sample Disable Register* (ADSDIS)**

Este registro permite habilitar solo las muestras deseadas. Con un cero lógico se habilitan y con un 1 lógico se deshabilitan. Por defecto todas las muestras están habilitadas. En esta aplicación se habilitó sólo la muestra 0.

Figura 34. Registro *ADC Sample Disable Register* (ADSDIS)

Base + $5	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	DS7	DS6	DS5	DS4	DS3	DS2	DS1	DS0
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MC56F8300 Peripheral User Manual, 2004.

- **Registro *ADC Status Register* (ADSTAT)**

Este registro proporciona información sobre el estado actual de módulo ADC. Para esta aplicación fue de particular interés monitorear el estado del bit CIP, que indica si una conversión está en progreso. Por defecto este bit se encuentra en el estado 0 lógico, cuando cambia a 1 lógico significa que hay una conversión en progreso y se debe esperar para poder guardar el resultado en el vector de muestras.

Figura 35. Registro *ADC Status Register* (ADSTAT)

Base + $6	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	CIP	0	0	0	EOSI	ZCI	LLMTI	HLMTI	RDY7	RDY6	RDY5	RDY4	RDY3	RDY2	RDY1	RDY0
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MC56F8300 Peripheral User Manual, 2004.

- **Registros ADC Result Registers (ADRSLT0–7)**

Hay ocho registros de resultados, que como indica su nombre, contienen los resultados de las conversiones del ADC. Si se ha utilizado el registro de offset, éste valor se restará del resultado de la conversión antes de almacenarlo en estos registros.

Figura 36. Registros *ADC Result Registers* (ADRSLT0–7)

Base + $9-10	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	SEXT	RSLT												0	0	0
Write		TEST_DATA														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MC56F8300 Peripheral User Manual, 2004.

El bit 15, SEXT, representa el signo del resultado. Un 1 lógico en este bit indica un resultado negativo.

- **Registros *ADC Offset Registers* (ADOFS0–7)**

Los registros de Offset sirven para corregir los resultados de las conversiones del ADC antes de guardarlos en los registros de resultados. En esta aplicación se utilizan para eliminar la componente DC de ajuste de la señal de 1,5V.

Figura 37. Registros *ADC Offset Registers* (ADOFS0–7)

Base + $21-28	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	OFFSET												0	0	0
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

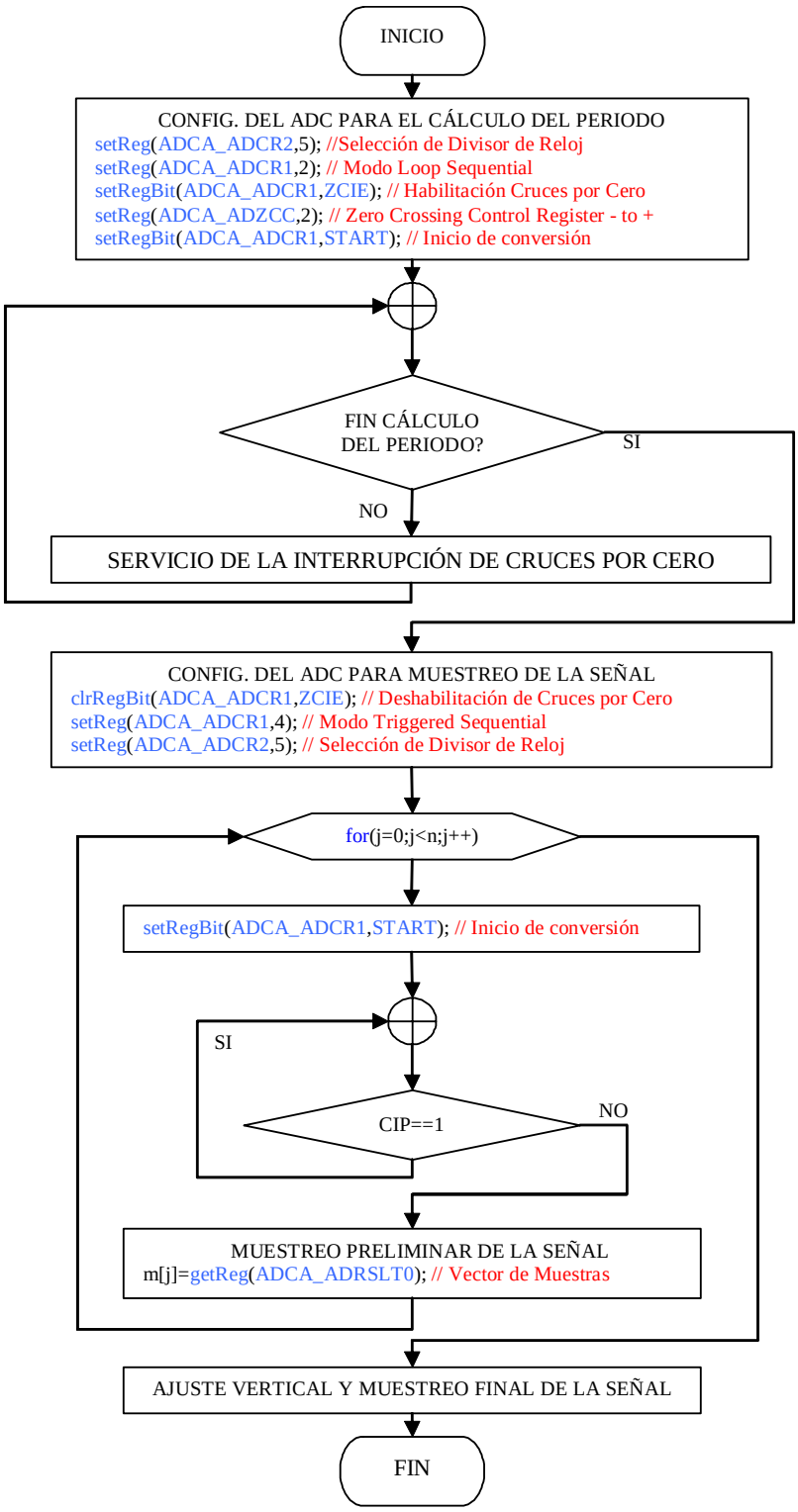
MC56F8300 Peripheral User Manual, 2004.

- **Configuración del ADC**

En esta sección se presenta el diagrama de flujo correspondiente a la etapa de adquisición de la señal mediante el ADC del DSP. Para este proceso se utilizaron dos configuraciones diferentes. Primero, para el cálculo del periodo, se habilitaron las interrupciones debidas al cruce por cero con el *bit* ZCIE (registro ADCR1), y se seleccionó el modo de *scan Loop Sequential* que permite tomar muestras de un canal hasta que se encuentre una bandera deshabilitada en el registro ADSDIS. Después, para tomar las muestras de la señal, se deshabilitaron las interrupciones, y se utilizó el modo de *scan Triggered Sequential*, en donde se toma una sola muestra y luego se almacena. Es importante mencionar que cada vez que se desee tomar una muestra en este modo se debe activar el disparo con un pulso del bit *START*. El servicio de la interrupción de cruces por cero se explicará posteriormente en la sección de medición de los parámetros de la señal.

Si se desea habilitar otros canales se recomienda utilizar los modos de *scan Loop Simultaneous* y *Triggered Simultaneous* para disminuir el tiempo de conversión. Estos modos de *scan* realizan conversiones simultáneas de dos canales y luego continúan con los dos siguientes hasta que se encuentre alguna bandera deshabilitada.

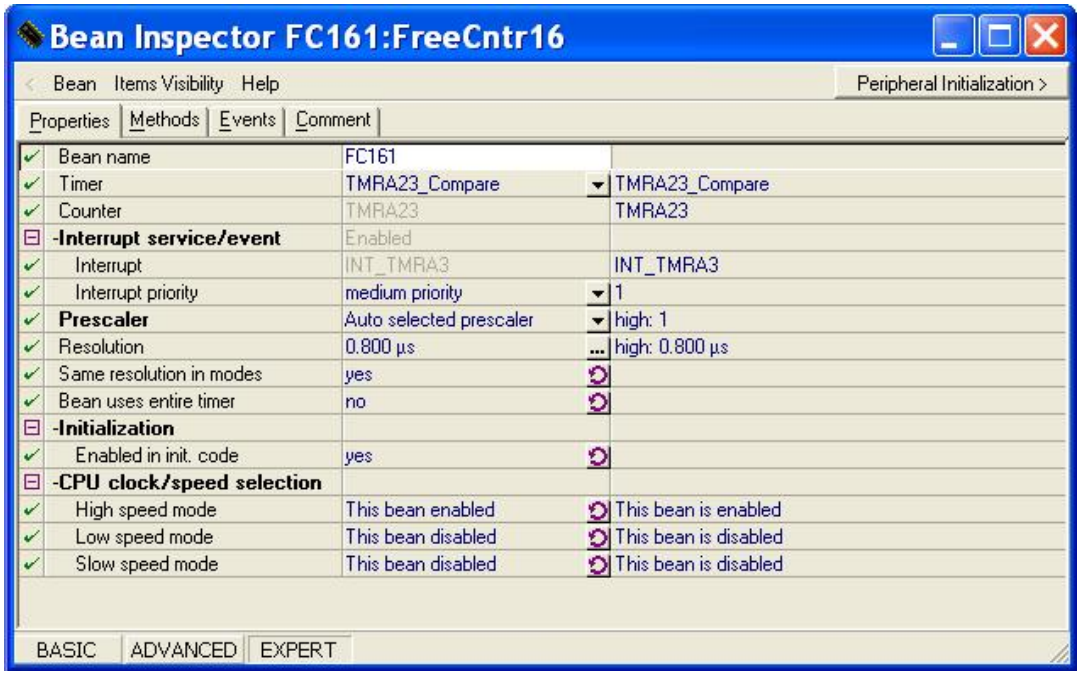
Figura 38. Diagrama de Flujo de Adquisición de la Señal


INICIO
CONFIG. DEL ADC PARA EL CÁLCULO DEL PERIODO
setReg(ADCA_ADCR2,5); //Selección de Divisor de Reloj
setReg(ADCA_ADCR1,2); // Modo Loop Sequential
setRegBit(ADCA_ADCR1,ZCIE); // Habilitación Cruces por Cero
setReg(ADCA_ADZCC,2); // Zero Crossing Control Register - to +
setRegBit(ADCA_ADCR1,START); // Inicio de conversión
FIN CÁLCULO DEL PERIODO?
SI
NO
SERVICIO DE LA INTERRUPCIÓN DE CRUCES POR CERO
CONFIG. DEL ADC PARA MUESTREO DE LA SEÑAL
clrRegBit(ADCA_ADCR1,ZCIE); // Deshabilitación de Cruces por Cero
setReg(ADCA_ADCR1,4); // Modo Triggered Sequential
setReg(ADCA_ADCR2,5); // Selección de Divisor de Reloj
for(j=0;j<n;j++)
setRegBit(ADCA_ADCR1,START); // Inicio de conversión
SI
CIP==1
NO
MUESTREO PRELIMINAR DE LA SEÑAL
m[j]=getReg(ADCA_ADRSLT0); // Vector de Muestras
AJUSTE VERTICAL Y MUESTREO FINAL DE LA SEÑAL
FIN

3.1.2. Timer (TMR)

El módulo *Timer* fue utilizado para el cálculo del periodo. Para su configuración se utilizó el *Bean Free Running 16-bit Counter* que se encuentra bajo la categoría *CPU Internal Peripherals/Timer/FreeCntr16*. Dentro de este *Bean* se encuentra el Método *GetTimeUS* que devuelve el tiempo en micro segundos desde el último Reset del contador, función que se realiza también mediante un método del mismo nombre.

Figura 39. Propiedades del *Bean FreeCntr16*



3.1.3. Pines de propósito general (GPIO)

Cómo se había mencionado en el primer capítulo, este módulo cuenta con 27 pines divididos en 3 puertos que se pueden configurar como GPIO. En la

siguiente tabla se muestra la configuración de los pines GPIO en el DSP 56F8323.

Tabla 9. Pines GPIO en el DSP 56F8323

Pin GPIO	Función Periférica	Pin 56F8323
GPIOA0	PWMA0	3
GPIOA1	PWMA1	4
GPIOA2	PWMA2/SSI	7
GPIOA3	PWMA3/MISO1	8
GPIOA4	PWMA4/MOSI1	9
GPIOA5	PWMA5/SCLK1	10
GPIOA6	FaultA0	13
GPIOA7	FaultA1	14
GPIOA8	FaultA2	15
GPIOA9	ISA0	16
GPIOA10	ISA1	18
GPIOA11	ISA2	19
GPIOB0	SS0/TXD1	21
GPIOB1	MISO0/RXD1	22
GPIOB2	MOSI0	24
GPIOB3	SCLK0	25
GPIOB4	HOME0/TA3	49
GPIOB5	INDEX0/TA2	50
GPIOB6	PHASEB0/TA1	51
GPIOB7	PHASEA0/TA0	52
GPIOC0	EXTAL	46
GPIOC1	XTAL	47
GPIOC2	CAN_RX	61

GPIOC3	CAN_TX	62
GPIOC4	TC3	63
GPIOC5	TC1/RXD0	64
GPIOC6	TC0/TXD0	1

- **Registro *Data Register* (DR)**

Este registro de lectura y escritura permite almacenar la información proveniente del pin o del bus IPBus. Si el pin se configura como salida, entonces el valor escrito en este registro se presentará en formato binario a la salida del puerto. Si el pin se configura como entrada, entonces se puede leer de este registro el valor presente a la entrada del puerto.

Figura 40. Registro *Data Register* (DR)

Base + $1	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	D							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MC56F8300 Peripheral User Manual, 2004.

En esta figura se asume un tamaño del puerto de 8 bits, se debe verificar el verdadero tamaño del puerto con la información presente en la Tabla.

- **Registro *Data Direction Register* (DDR)**

Este registro de lectura y escritura configura el pin como entrada o salida cuando el grupo de bits PE del registro PER se encuentre en un nivel lógico bajo. Cuando DD tiene un valor igual a 0, el pin es una entrada, cuando DD tiene un valor igual a 1, el pin está configurado como salida.

Figura 41. Registro Data Direction Register (DDR)

Base + $2	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	DD							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fuente: MC56F8300 Peripheral User Manual, 2004.

En esta figura se asume un tamaño del puerto de 8 bits, se debe verificar el verdadero tamaño del puerto con la información presente en la Tabla.

- **Registro *Peripheral Enable Register* (PER)**

Este registro de lectura y escritura es el que determina la configuración del GPIO. Cuando el valor de PE es 1, el módulo GPIO está configurado en modo normal. En este modo, el módulo GPIO está controlado por un periférico (ver Tabla). Cuando el valor de PE es 0, el módulo GPIO está configurado para trabajar como GPIO y no como pin dedicado. En este modo, el GPIO permite escribir en el puerto, si éste está configurado como salida, o permite leer información del puerto si éste esta configurado como entrada.

Figura 42. Registro *Peripheral Enable Register* (PER)

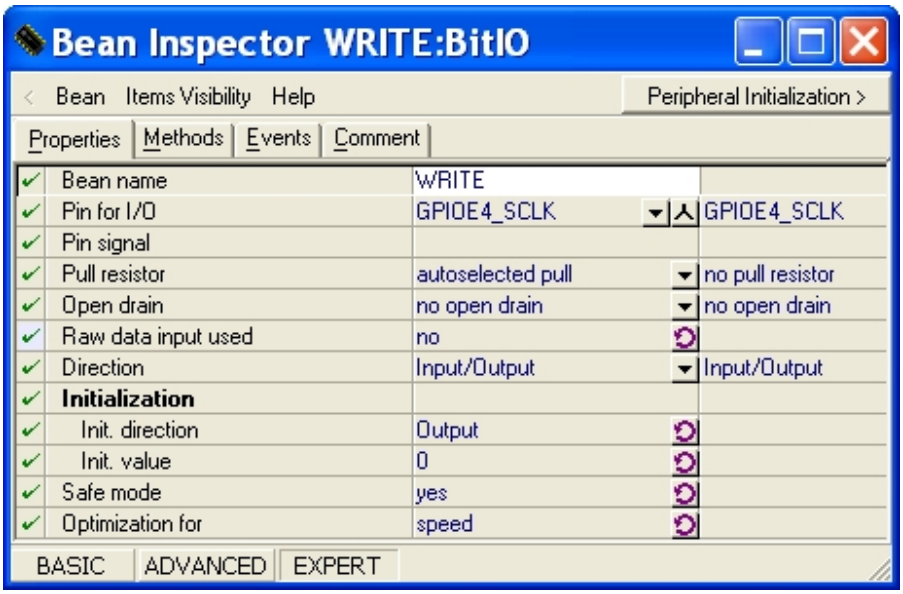
Base + $3	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0	0	0	0	0	0	0	0	PE							
Write																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Fuente: MC56F8300 Peripheral User Manual, 2004.

- **Configuración de GPIO**

Para configurar los pines de propósito general se utilizaron dos métodos. En primera instancia se configuraron escribiendo directamente a los registros de este módulo expuestos en las secciones anteriores, y para configurar aquellos pines con funciones independientes se utilizó el *Bean BitIO*, que se encuentra bajo la categoría *CPU Internal Peripherals/Port I/O*. En la siguiente figura se pueden apreciar las posibilidades de configuración que ofrece este *Bean*.

Figura 43. Propiedades del *Bean BitIO*



Por medio de este *Bean* se puede seleccionar el pin del DSP que se quiere destinar para propósito general. Éste se puede configurar ya sea como salida o como entrada y además se puede seleccionar su valor inicial, aunque por supuesto éste valor se puede modificar dentro del programa

utilizando alguno de los métodos destinados para este propósito, como lo son *SetVal*, *PutVal* y *ClrVal*.

Como se explicó anteriormente, los pines de propósito general también se pueden configurar escribiendo directamente a los registros de éste módulo. En el siguiente diagrama de flujo se puede apreciar un ejemplo de cómo configurar un puerto como GPIO.

Figura 44. Diagrama de flujo de configuración de un puerto como GPIO

INICIO
setReg(GPIO_B_DDR,0xFF); // Configura el Puerto B como salida
setReg(GPIO_B_PER,0x00); // Habilita el Puerto B como GPIO
setReg(GPIO_B_DR,0xFF); // Escribe el valor 11111111 a la salida del puerto
FIN

La siguiente tabla proporciona información sobre cómo fueron configurados los pines GPIO en el prototipo final, así como su función dentro de la aplicación.

Tabla 10. Configuración de los pines GPIO en el prototipo

GPIO	**Pin 56F8323**	**Función**
GPIOA0	3	LCD_D0
GPIOA1	4	LCD_D1
GPIOA2	7	LCD_D2
GPIOA3	8	LCD_D3

GPIOA4	9	LCD_D4
GPIOA5	10	LCD_D5
GPIOA6	13	LCD_D6
GPIOA7	14	LCD_D7
GPIOB0	21	START
GPIOB1	22	LCD WR
GPIOB2	24	LCD A0
GPIOB3	25	Switch C_{paso} S1
GPIOB4	49	Switch Atenuación S1
GPIOB5	50	Switch Atenuación S2
GPIOB6	51	Switch Atenuación S3
GPIOB7	52	Switch Atenuación S4

3.2. FILTRO DIGITAL

Para lograr un filtrado más estricto de frecuencias no deseadas, en este caso frecuencias superiores a 20KHz fue necesaria la implementación de un filtro digital pasabajas que ayude a mejorar la respuesta en frecuencia del sistema.

Para nuestra implementación en especial se eligió el diseño de un filtro FIR (*Finite Impulse Response*) debido a la necesidad de una fase lineal. En general un filtro FIR se puede describir mediante la siguiente expresión:

$$y(n) = b_0 \cdot x(n) + b_1 \cdot x(n-1) + ... + b_{M-1} x(n-M+1)$$

$$y(n) = \sum_{k=0}^{M-1} b_k \cdot x[n-k] \qquad (7)$$

Donde *x(n)* representa la entrada del filtro, *M* la longitud del filtro, *y(n)* la salida del filtro y b_k los coeficientes del filtro.

Los coeficientes del filtro b_k se pueden deducir a partir de la respuesta del filtro *h(n)*, expresada por la ecuación (7), a una entrada impulso δ*(n)* de la siguiente manera:

$$y(n) = \sum_{k=0}^{M} b_k \cdot \delta(n-k) = h(n) \quad (8)$$

Donde el impulso cumple con las siguientes condiciones:

$$\delta(n-k) = \begin{cases} 0 & \text{para n} \neq \text{k} \\ 1 & \text{para n} = \text{k} \end{cases}$$

Para que el filtro tenga fase lineal su respuesta a la entrada impulso debe cumplir con la siguiente restricción:

$$h(n) = \pm h \cdot (M-n) \quad (9)$$

De la ecuación (8) se deduce que los coeficientes b_k son muestras de la respuesta al impulso δ*(n)* [Herrera y Florez, 2004], es decir:

$$\begin{aligned} b_0 &= h(0) \\ b_1 &= h(1) \\ b_2 &= h(2) \\ &\vdots \\ b_k &= h(k) \end{aligned}$$

Para el diseño de los filtros FIR de fase lineal y para el cálculo de sus coeficientes existen dos métodos básicos que son: el diseño de filtros FIR usando ventanas y el diseño de filtro FIR mediante el método de muestreo de frecuencia. Particularmente para este prototipo, el método de diseño del filtro

FIR escogido fue el primero debido a que las funciones ventana eliminan los rizados en el límite de la banda de paso y reducen los lóbulos laterales en el dominio de la frecuencia, y aunque su región de transición es grande, este problema es solucionado con el aumento de la longitud del filtro.

Dentro de las funciones ventana con una caída a cero suavizada que garanticen una disminución de oscilaciones dentro de la banda de paso y transición, existen varias formas entre las que se encuentran las ventanas Hamming, Hanning, Blackman y Kaiser. La ventana seleccionada fue Kaiser debido a que permite una mayor atenuación, de -98 dB en comparación con las demás ventanas mencionadas [Herrera y Florez, 2004].
La ventana Kaiser esta definida por la ecuación:

$$w(n) = \begin{cases} \dfrac{Io\left[\beta \cdot \left(1 - \left[\left(n-\alpha\right)/\alpha\right]^2\right)^{1/2}\right]}{Io \cdot \beta}, & 0 \le n \le M \\ 0, \quad \text{en el resto} \end{cases} \quad (10)$$

Donde α=*M/2* e Io(-) es la función de Bessel modificada de primera especie [Oppenheim & Schafer, 2000]. La longitud y la forma de la ventana son modificadas por medio de las variables *M* y *β* respectivamente. Para determinar el valor de la variable *β* que Kaiser estableció empíricamente en función de la atenuación deseada (A), se utilizan las siguientes ecuaciones:
lead

$$\beta = \begin{cases} 0.1102 \cdot (A - 8.7), & A > 50 \\ 0.5842 \cdot (A-21)^{0.4} + 0.07886 \cdot (A-21), & 21 \le A \le 50 \\ 0, & A < 21 \end{cases}$$

Adicionalmente Kaiser calculó el valor de *M* a partir de los valores de A y del ancho de la banda de transición $\Delta\omega$ por medio de la siguiente ecuación:

$$M = \frac{A-8}{2.285 \cdot \Delta\omega} \quad (11)$$

Donde $\Delta\omega$ esta definida por la diferencia en radianes entre la frecuencia de corte de la banda de atenuación ω_s y la frecuencia de corte de la banda de paso ω_p, es decir, $\Delta\omega = \omega_s - \omega_p$. La Ecuación (11) permite predecir el valor de *M* con una precisión de ±2 para un amplio margen de valores de $\Delta\omega$ y *A.* [Oppenheim & Schafer, 2000].

Para obtener la respuesta en frecuencia del filtro FIR pasabajas con ventana Kaiser y sus coeficientes se utilizo el programa MATLAB 6.5. A partir de las ecuaciones anteriormente descritas y teniendo en cuenta que la atenuación deseada es de 98 [dB], el ancho de la banda de transmisión $\Delta\omega$ elegido es de 0.09846 [rad], la frecuencia de corte escogida es de 21KHz y la frecuencia de sobre muestreo es de 400kHz, se obtuvieron los siguientes parámetros necesarios para la realización de las simulaciones:

- $\beta = 0.1102 \cdot (98 - 8.7) = 9.84086$
- $M = 400$
- $\Delta\omega = \frac{A-8}{2.285 \cdot \Delta\omega} = \frac{98-8}{2.285 \cdot 400} = 0.09846$

La rutina implementada para obtener los coeficientes b_k y las gráficas de simulación se puede encontrar en el Anexo B. En las Figuras 45 y 46 se verifican los resultados de dichas simulaciones, el barrido de frecuencias utilizadas en su realización fue hasta la mitad de la frecuencia de sobre muestreo que equivale a un intervalo en radianes entre 0 y π debido a requerimientos en el uso de comandos en el programa MATLAB.

En el software final del prototipo este filtro FIR no fue incluido debido a que almacenar los coeficientes b_k en la memoria flash del DSP excedía la capacidad de la licencia de CodeWarrior utilizada, sin embargo se deja a disposición su futura implementación.

Figura 45. Respuesta en frecuencia del filtro FIR pasabajas diseñado.

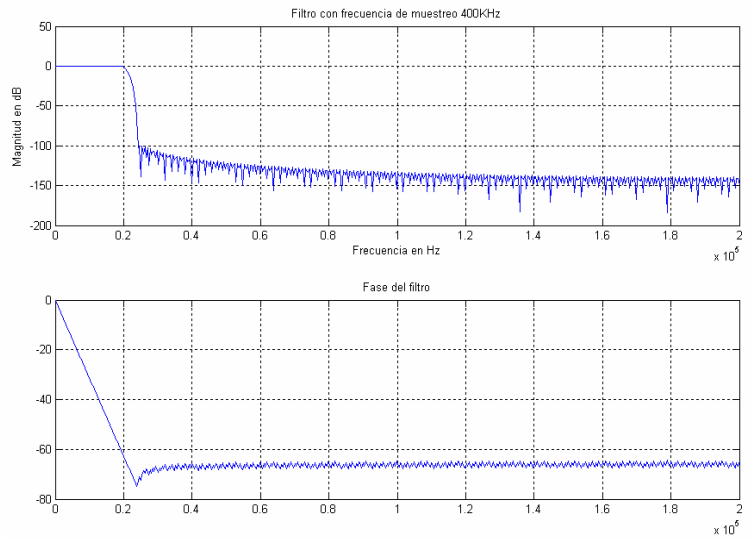
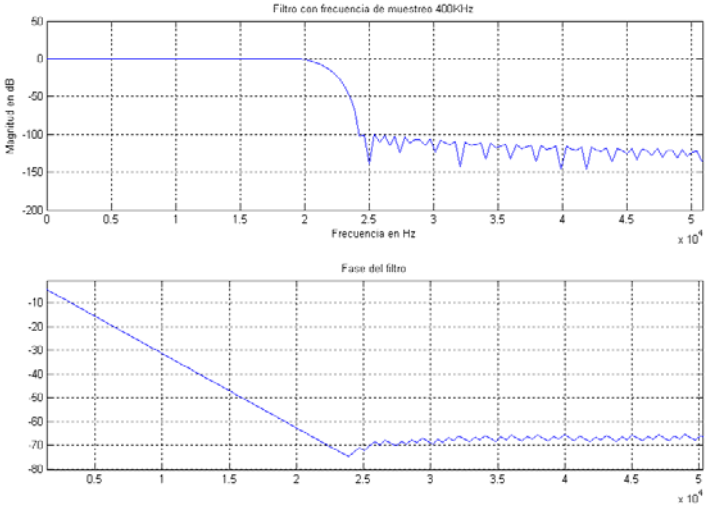


Figura 46. Detalle de la respuesta en frecuencia del filtro FIR.



3.3. MEDICIÓN DE LOS PARÁMETROS DE LA SEÑAL

Los métodos de medición de parámetros en los osciloscopios están basados en los estándares del IEC (*International Electrotechnical Commission*). Estos métodos utilizados para la medición de parámetros de señales periódicas en el dominio del tiempo se basan en cruces por un nivel de referencia.

Los estándares del IEC en los cuáles están basados los algoritmos de medición son: *Pulse Techniques and Apparatus—Part 1: Pulse Terms and Definitions*, IEC 60 469 Ed. 2.0b, y, *Pulse Techniques and Apparatus—Part 1: Pulse Measurement and Analysis*, IEC 60 469 Ed. 2.0b. Estos algoritmos son relativamente simples y requieren una capacidad de procesamiento y memoria moderada.

Los parámetros de las señales son calculados diez veces según criterio empírico y promediados antes de ser mostrados en la pantalla gráfica LCD para reducir el error en los valores medidos y evitar que al refrescarse la pantalla se limite la visualización por parte del usuario.

3.3.1. Cálculo del periodo

Para explicar el método de medición del periodo de una señal primero se debe ampliar un poco la definición de periodo expuesta en el primer capítulo. El periodo P de una señal verdaderamente periódica s(t) está definido como s(t)=s(t+P), para cualquier t. Sin embargo, para una medición práctica este requerimiento es demasiado riguroso, en lugar de esto se dice que el periodo P de una señal periódica está definido tal que s(t) es lo más similar a s(t+P).

Como se había mencionado antes, el método elegido consiste en determinar los cruces por el nivel de referencia que hace la señal. Este nivel, MidRef, normalmente se ajusta al 50% del valor pico a pico, aunque la mayoría de los

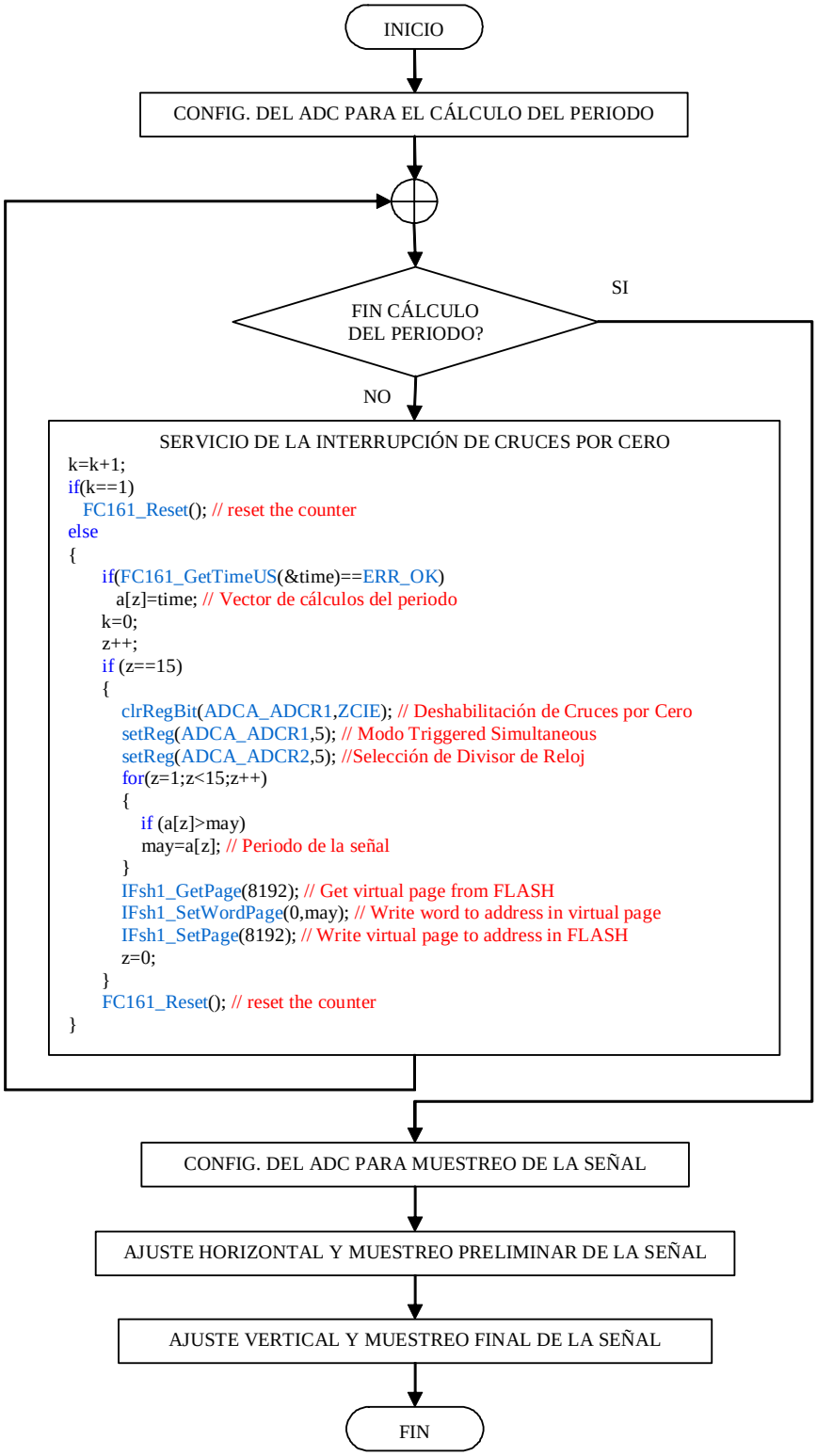
osciloscopios permiten al usuario ajustar este nivel en las opciones de disparo. El instante en el que la forma de onda cruza por MidRef se define como t_m. Las pendientes en los cruces deben cambiar de polaridad y por consiguiente el periodo queda determinado como:

$$P = t_3 - t_1$$

Utilizando los registros del Conversor Análogo Digital expuestos en la sección 3.1.1 se puede habilitar una interrupción cuando la señal cruce por un nivel de referencia que se ajusta mediante los registros de Offset dispuestos para cada canal. El primer paso para la medición del periodo de la señal consiste en configurar el ADC para este propósito. Primero se habilitan las interrupciones de cruces por cero con el bit *ZCIE (Zero Crossing Interrupt Enable)* del registro del control *ADCR1*, seguidamente se ajusta el registro de offset *(ADC Offset Registers (ADOFS0–7))* a 1.6V, que es el nivel utilizado externamente en el acondicionamiento de la señal y que permite ubicarla dentro del rango de voltaje permitido por el ADC, de 0V a 3.3V. Posteriormente en el registro de control de cruces por cero, *ADC Zero Crossing Control Register (ADZCC)*, se selecciona la opción de monitoreo de cambio de signo de negativo a positivo. Como es posible que la señal tenga una componente de DC no conocida, entonces ésta se elimina haciendo circular la señal por un condensador de paso. Finalmente se da inicio a la conversión en modo *Loop Sequential*.

El diagrama de flujo correspondiente al cálculo del periodo se muestra a continuación.

Figura 47. Diagrama de flujo para el cálculo del periodo.



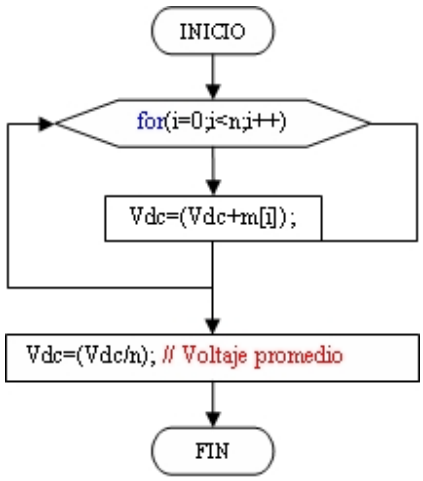
Es necesario aclarar que para llevar el valor del periodo, calculado en el servicio de la interrupción, al programa principal fue necesario escribir directamente en al memoria FLASH del DSP. Para esto se utilizó el Bean *IntFLASH* que se encuentra bajo la categoría *CPU Internal Peripherals/Memory/IntFLASH*. El método *GetPage* permite utilizar una hoja virtual en la memoria FLASH, el método *SetWordPage* permite escribir una palabra en la página virtual y finalmente el método *SetPage* escribe la página virtual en la dirección especificada de la memoria FLASH. Luego, para recuperar el valor dentro del programa principal se utilizan los métodos *GetPage* y *GetWordPage*. El primero permite utilizar una página virtual en la memoria FLASH, y el segundo permite recuperar una palabra de la página virtual.

3.3.2. Cálculo del valor promedio

De acuerdo a la definición expuesta en la sección 1.1 el valor promedio es simplemente la media de la magnitud. En esta aplicación la magnitud está representada por n valores discretos, entonces la sumatoria se extenderá sobre el número de muestras capturadas o en el caso de una señal periódica sobre un número entero de periodos de la señal.

El código utilizado para el cálculo del valor promedio se muestra en la siguiente figura, donde m[i] corresponde al vector de muestras y n al número de muestras.

Figura 48. Diagrama de flujo del cálculo del valor promedio



3.3.3. Cálculo del valor RMS

El valor RMS o valor efectivo en una definición menos formal se puede considerar como "una medida de la efectividad de una fuente de voltaje para entregar potencia a una carga resistiva" [Kemmerly, Hayt, JR., 1993].

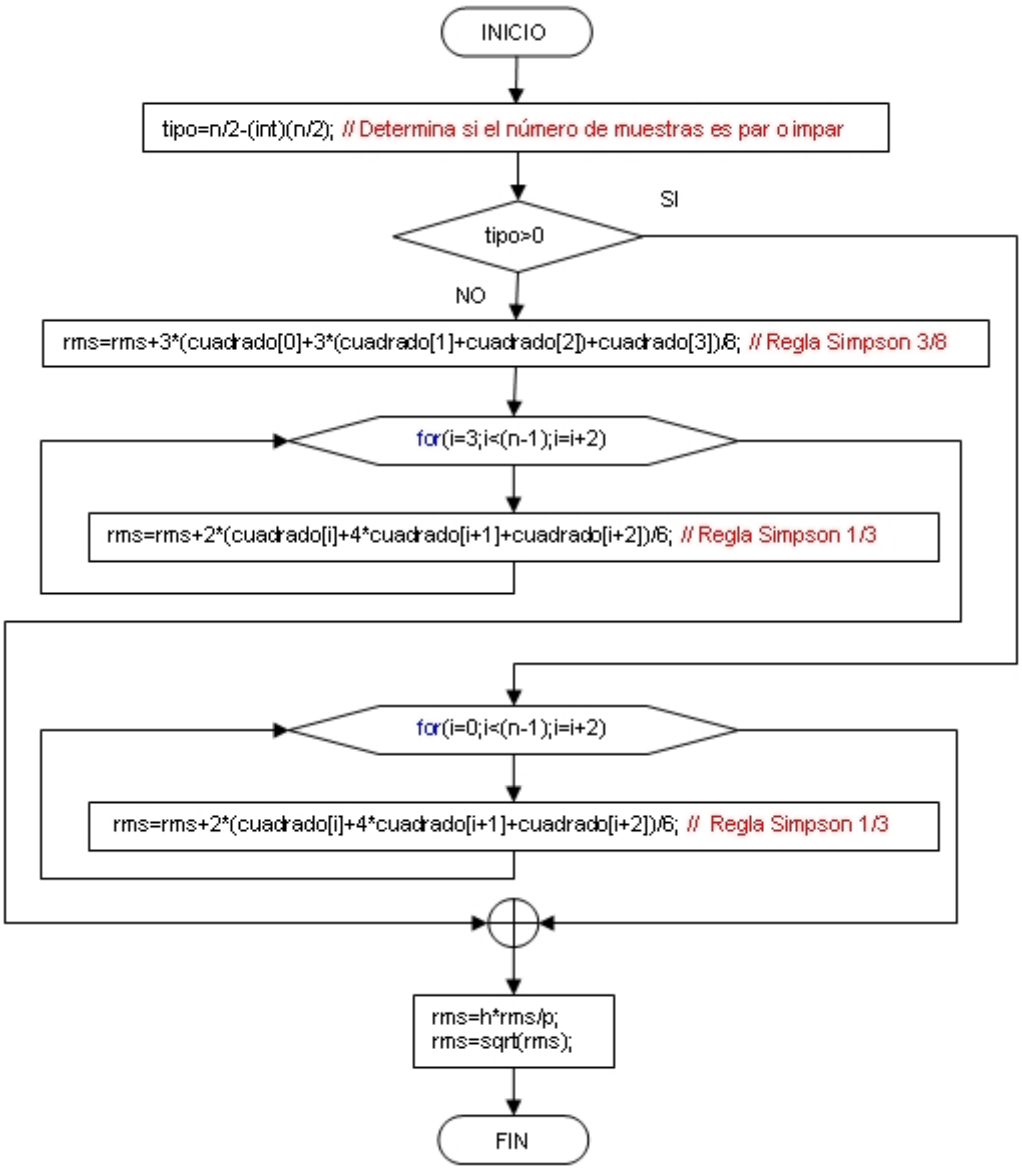
$$V_{RMS} = \sqrt{\frac{1}{T}\int_0^T v^2 dt}$$

Para calcular este parámetro se utilizan las Reglas de *Simpson*, dos métodos de integración numéricos polinomiales que permiten obtener una estimación más exacta de la integral. La Regla de *Simpson* 1/3 utiliza una interpolación polinomial de segundo orden para estimar el valor de la integral. Este método sólo se puede implementar si el número de muestras es impar. Sí el número de muestras es par, se debe recurrir a la Regla de *Simpson* 3/8 donde se utiliza un polinomio de tercer orden y se requieren como mínimo cuatro muestras. El algoritmo implementado hace uso de los dos métodos dependiendo del número de muestras. Cuando se tiene un número impar de muestras se aplica la Regla de *Simpson* 1/3, y cuando se tiene un número

par de muestras se aplica la Regla de *Simpson* 3/8 a los primeros 3 segmentos y la Regla 1/3 a los segmentos restantes.

El algoritmo para este método se muestra en la siguiente figura, donde n representa el número de muestras, cuadrado[i] el vector de muestras al cuadrado, h el tamaño de paso y p el periodo de la señal.

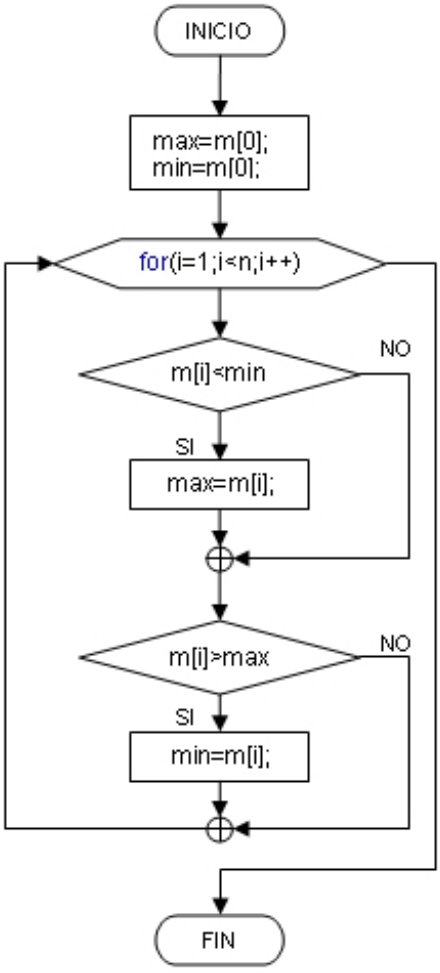
Figura 49. Diagrama de flujo del cálculo del valor RMS



3.3.4. Cálculo del valor pico a pico

El método para calcular el valor pico a pico consiste en detectar cuál es la muestra de mayor magnitud y la muestra de menor magnitud y tomar la diferencia algebraica. El algoritmo implementado se muestra a continuación, donde max representa la muestra de mayor magnitud, min la muestra de menor magnitud y n el número de muestras.

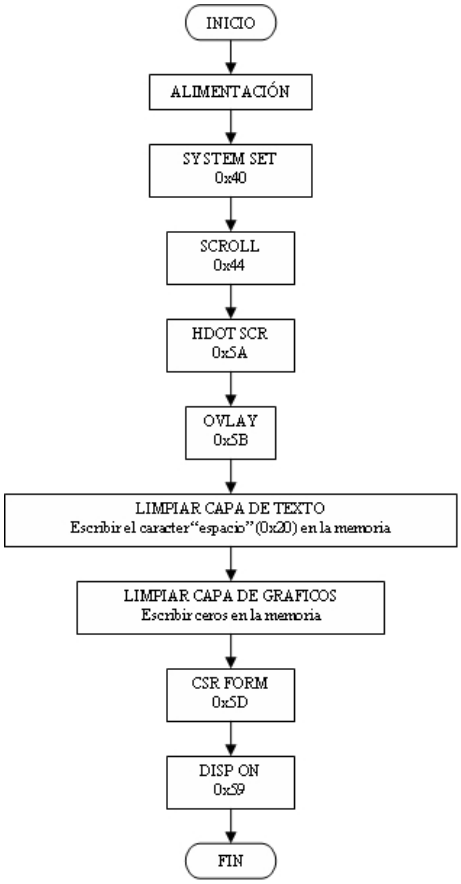
Figura 50. Diagrama de flujo del cálculo del valor pico a pico



3.4. INICIALIZACIÓN Y MANEJO DE LA PANTALLA GRÁFICA LCD

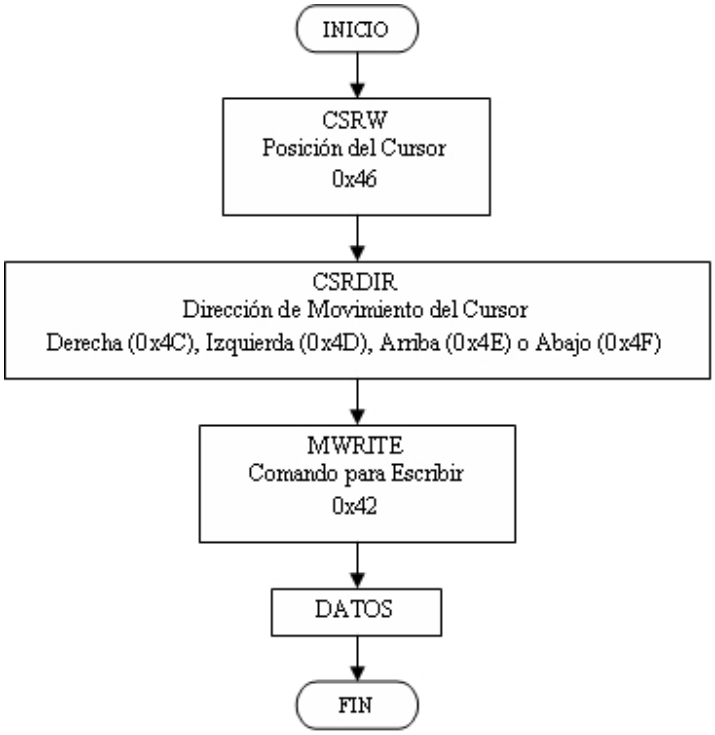
La inicialización de la pantalla gráfica LCD consiste en enviar al controlador de la pantalla los comandos y datos necesarios para configurarla y de esta forma determinar sus características de funcionamiento. En el capítulo 2 se presentaron las instrucciones necesarias para la inicialización de la pantalla así como su configuración final. En la siguiente figura se muestra un diagrama de flujo en el que se enumeran los pasos que se deben seguir para inicializar la pantalla correctamente. Es necesario aclarar que aquí sólo se presentan los comandos, más no los datos que se deben enviar después de cada comando para configurar la pantalla.

Figura 51. Diagrama de flujo inicialización LCD



Una vez se ha inicializado la pantalla se puede proceder a enviar datos a ella para su posterior visualización. La escritura de datos en la pantalla se hace mediante una serie de instrucciones que determinan la posición del cursor en la pantalla (capa de texto o gráfica), la dirección de movimiento del cursor (arriba, abajo, izquierda o derecha) y finalmente el comando de escritura seguido del código ASCII de los caracteres (capa de texto) o de los puntos del gráfico (capa de gráficos).

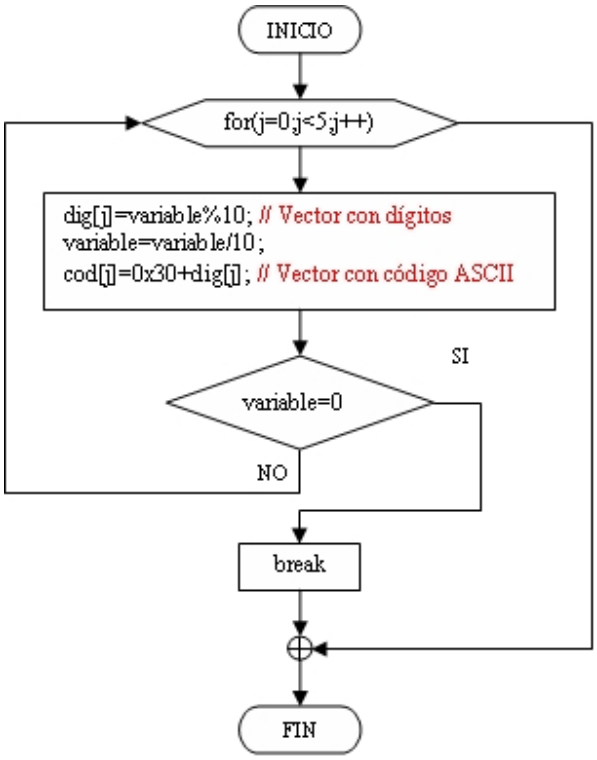
Figura 52. Diagrama de flujo escritura de datos en la LCD



En esta aplicación es necesario escribir en pantalla los valores de los parámetros calculados. Para este propósito es necesario identificar cada dígito del número y posteriormente enviar el código ASCII de este dígito a la pantalla. En el siguiente diagrama de flujo se puede apreciar el método

utilizado para separar la variable en dígitos y posteriormente guardar su código ASCII correspondiente en un vector para su visualización. El método consiste en hacer una división por diez utilizando el operador % (resto de la división entera). Este operador permite recuperar el último digito de la variable. Posteriormente se hace una división normal por diez para ir descartando de la variable los dígitos ya almacenados. Para cada dígito del número se debe calcular el código ASCII; en este estándar los números tienen un código que va desde el 0x30 para el cero hasta el 0x39 para el nueve, entonces luego de obtener el dígito se debe sumar 0x30 para obtener su correspondiente código ASCII. La sentencia IF permite identificar si el número ya fue separado completamente en sus dígitos.

Figura 53. Diagrama de flujo código ASCII de variables



4. PRUEBAS Y RESULTADOS OBTENIDOS

Para el análisis del desempeño del prototipo *DIGISCOPE* se realizaron diferentes pruebas con el fin ajustar la visualización de las gráficas en la pantalla HG25504NG-01 de Hyundai y comparar los valores de frecuencia, voltaje pico a pico, valor RMS y valor promedio obtenidos con los osciloscopios digitales TEKTRONIX TDS 210 y FLUKE 105 Series II.

4.1. PRUEBAS EN LA PANTALLA GRÁFICA HG25504NG-01 DE HYUNDAI.

Al energizar el prototipo *DIGISCOPE* se observa en la pantalla LCD la presentación del dispositivo que contiene el logo de la Universidad Industrial de Santander, el nombre del prototipo, el nombre de los autores, el grupo de investigación al cual pertenece este trabajo y la sigla E3T, que identifica a la Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones.

Figura 54. Portada del prototipo *DIGISCOPE.*

Posteriormente el usuario podrá observar en la pantalla LCD un marco que le indica que puede conectar la señal.

Figura 55. Visualización del marco en la pantalla LCD

Finalmente al conectar la señal el usuario podrá visualizar la forma de la señal de entrada, la frecuencia de la señal, su valor pico a pico, su valor promedio y su valor RMS.

4.2. PRUEBAS SOBRE LAS MEDICIONES DE PARÁMETROS CON UNA SEÑAL DE VOLTAJE PICO A PICO CONSTANTE.

Con el propósito de realizar diferentes mediciones en los valores de frecuencia, voltaje pico a pico, valor RMS y valor promedio en el prototipo *DIGISCOPE* y compararlas con los resultados obtenidos en los osciloscopios digitales TEKTRONIX TDS 210 y FLUKE 105 Series II, se aplicó una señal senoidal con un valor de voltaje pico a pico de 2V por medio de un generador de señales y se realizaron variaciones de frecuencia entre un rango de 1KHz

y 20KHz en cada uno de los dispositivos. Los rangos de valores obtenidos se mencionan a continuación:

Tabla 11. Mediciones de frecuencia con un voltaje pico a pico de 2V.

TEKTRONIX TDS 210 (KHz)	***DIGISCOPE* (KHz)**	**FLUKE 105 SERIES II (KHz)**
19,96 – 20	19,8 – 20,4	19,8 – 20,1
18,98 – 19,01	18,8 – 19,6	18,9 – 19,2
17,92 – 18,02	17,85	17,8 – 18,1
16,98 – 17,01	16,9 – 17,24	16,8 – 17,1
15,97 – 16	15,87	15,8 – 16
14,97 – 15,06	14,7 – 15,15	14,8 – 15
13,97 - 14,01	13,98 - 14,08	13,8 – 14
12,95 – 13	12,82 – 13,15	12,8 – 13
11,96 – 12,02	11,76 – 12,04	11,9 – 12
11 – 11,03	10,86 - 11,1	10,9 – 11
9,98 – 10,03	10 – 10,1	10
8,99 – 9,04	8,92 – 9,09	8,92 – 9,09
7,98 – 8,01	7,87 – 8,26	7,93 – 8,06
6,98 – 7	6,99 – 7,09	6,94 – 7,04
5,99 – 6,01	5,98 – 6,06	5,95 – 6,02
4,97 – 5,01	4,97 – 5,09	4,9 – 5
3,98 – 4,00	3,99 – 4,08	3,96 – 4,03
2,99 – 3,01	3,09 – 3,1	2,87 – 3,01
2,02 – 2,03	2,008 – 2,03	2,01 – 2,04
1,012 – 1,015	1,01 – 1,06	1 – 1,02

Los siguientes valores de voltaje pico a pico, voltajes promedio y RMS se midieron para cada uno de los valores de frecuencia mencionados en la tabla anterior.

Tabla 12. Mediciones de voltaje pico a pico con respecto a la variación de frecuencia.

TEKTRONIX TDS 210 (V)	***DIGISCOPE* (V)**	**FLUKE 105 SERIES II (V)**
2 - 2,04	1,95 – 1,97	1,90 – 1,95
2 - 2,04	1,95	1,90 – 1,92
2 - 2,04	1,98 – 2	1,90 – 1,92
2 - 2,04	1,96 - 2	1,90 – 1,92
2 - 2,04	1,99 – 2,02	1,90 – 1,94
2 - 2,04	1,99 – 2,02	1,90 – 1,92
2 - 2,04	1,96 – 1,99	1,90 – 1,92
2 - 2,04	1,93 – 1,97	1,90 – 1,92
2,02 – 2,04	1,96 – 1,99	1,96 – 1,98
2 – 2,02	1,95 – 2,05	1,94 – 1,98
2,02 – 2,04	1,96 - 2	1,96 – 1,98
2,02 – 2,04	1,93 – 1,97	1,92 – 1,94
2,04 – 2,06	2,02 – 2,04	1,98 – 2
2,04 – 2,06	2,002 – 2,02	2
2,04 – 2,06	1,98 – 2,019	2 – 2,02
2,04 – 2,08	2,001 – 2,05	2
2,04 – 2,06	1,98 – 2,1	1,98 – 2,02
2,02 – 2,04	1,98 – 2,03	2 – 2,02
2,02 – 2,04	2,06 – 2,1	2 – 2,02
2 – 2,02	1,85 – 2,12	2 – 2,02

Tabla 13. Mediciones de valor promedio con respecto a la variación de frecuencia.

TEKTRONIX TDS 210 (mV)	***DIGISCOPE* (mV)**	**FLUKE 105 SERIES II (mV)**
(-23,5) – (-22)	52 – 71	80
11,0 – 12,8	50 – 65	80 – 100
25,2 – 7,3	44 – 64	80
11,5 – 14,5	56 – 58	40
(-25) – (-23,1)	66 – 85	-20 – 0
(- 69,9) – (-65,1)	59 – 79	-60
(- 87.6) – (-85.8)	-52 – 97	(-100) – (-80)
(-73.7) – (-70.3)	-75 – 100	-160
(-23.9) – (-21.3)	-45 – 100	(-140) – (-120)
(-30.1) – (-28.4)	-55 – 96	-120 – 100
(-23.2) – (-21.8)	-22 – 70	20
(-44.9) – (-42.2)	-97 – 50	-80

(-49.7) – (-46.9)	-87 – 320	80
(-52) – (-49.4)	-81 – 30	40
(-43.3) – (-41.7)	-83 – 103	-80
(-32.9) – (-29.9)	-85 – 60	(-40) – (-20)
(-40.7) – (-37.8)	-97 – 44	60 – 80
(-53.2) – (-48.8)	-84 – 105	(-100) – (-80)
(-23.8) – (-21.8)	-15 – 330	-100
(-6.69) – (-5.31)	-129 – 84	-40

Tabla 14. Mediciones de valor RMS con respecto a la variación de frecuencia.

TEKTRONIX TDS 210 (mV)	***DIGISCOPE* (mV)**	**FLUKE 105 SERIES II (mV)**
675 – 679	646 – 655	660
674 – 800	687 - 691	660
675 – 678	669 - 671	680
674 – 676	664 – 670	680
674 – 677	677 – 687	660
673 – 679	678 – 684	640
672 – 676	640 – 690	660
672 – 677	645 – 687	640
690 – 692	669 – 682	680
690 – 693	670 – 707	660
689 – 694	674 – 709	700
690 – 694	675 – 707	660
703 – 707	688 – 731	680
704 – 706	678 – 717	700
704 – 707	679 – 715	680
701 – 703	673 – 704	680
701 – 703	669 – 711	680
700 – 702	667 – 726	680
699 – 702	689 – 704	700
699 – 701	689 – 726	680

Para comparar los valores obtenidos se seleccionó como referencia las mediciones mostradas por el osciloscopio digital TEKTRONIX TDS 210 y se calcularon los porcentajes de error producidos por el osciloscopio FLUKE 105

SERIES II y el Prototipo *DIGISCOPE* como muestran las siguientes tablas y figuras:

Tabla 15. Porcentajes de error en la medición de valores de frecuencia.

Promedio de la frecuencia medida por TEKTRONIX TDS 210 (KHz.)	**ERROR EN DIGISCOPE (%)**	**ERROR EN FLUKE 105 SERIES II (%)**
19,98	0,600600601	0,15015015
18,995	1,079231377	0,289549882
17,97	0,667779633	0,111296605
16,995	0,441306267	0,26478376
15,985	0,71942446	0,531748514
15,015	0,599400599	0,765900766
13,99	0,285918513	0,643316655
12,975	0,077071291	0,578034682
11,99	0,750625521	0,333611343
11,015	0,317748525	0,590104403
10,005	0,449775112	0,049975012
9,015	0,110926234	0,110926234
7,995	0,875547217	0
6,99	0,715307582	0
6	0,333333333	0,25
4,99	0,801603206	0,801603206
3,99	1,127819549	0,125313283
3	3,166666667	2
2,025	0,296296296	0
1,0135	2,121361618	0,345337938

Figura 56. Error en la medición de valores de frecuencia.

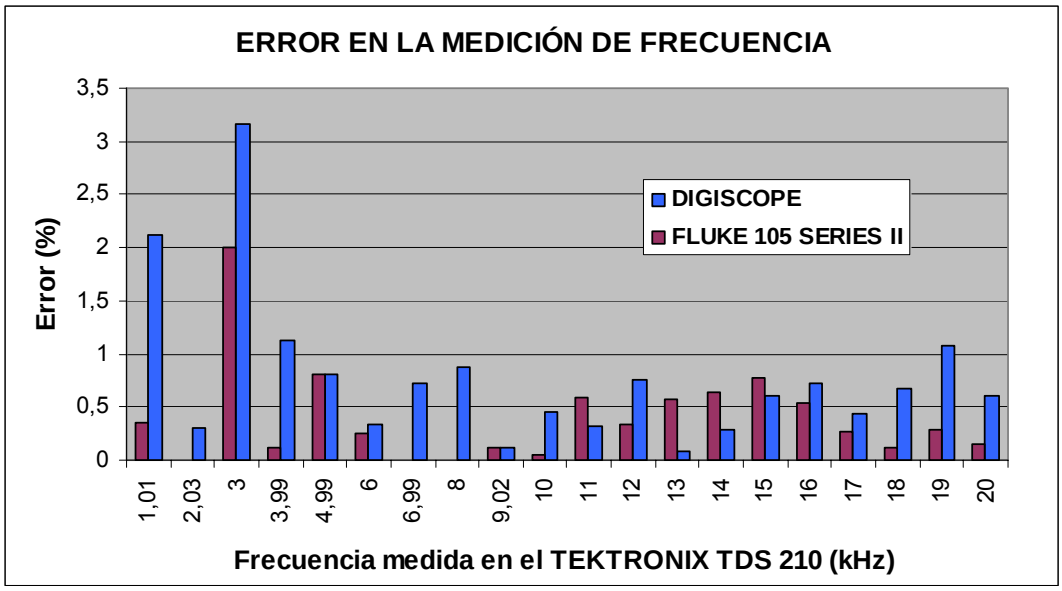


Tabla 16. Porcentajes de error en la medición de valores de voltaje pico a pico.

Promedio de la frecuencia medida por TEKTRONIX TDS 210 (KHz.)	Promedio del Vpp medido por TEKTRONIX TDS 210 (V)	ERROR EN DIGISCOPE (%)	ERROR EN FLUKE 105 SERIES II (%)
19,98	2,02	2,97	4,702970297
18,995	2,02	3,47	5,445544554
17,97	2,02	1,49	5,445544554
16,995	2,02	1,98	5,445544554
15,985	2,02	0,74	4,95049505
15,015	2,02	0,74	5,445544554
13,99	2,02	2,23	5,445544554
12,975	2,02	3,47	5,445544554
11,99	2,03	2,71	2,955665025
11,015	2,01	0,50	2,487562189
10,005	2,03	2,46	2,955665025
9,015	2,03	3,94	4,926108374
7,995	2,05	0,98	2,926829268
6,99	2,05	1,90	2,43902439
6	2,05	2,93	1,951219512

4,99	2,06	1,70	2,912621359
3,99	2,05	0,49	2,43902439
3	2,03	1,23	0,985221675
2,025	2,03	2,46	0,985221675
1,0135	2,01	1,24	0

Figura 57. Error en la medición de valores de voltaje pico a pico.

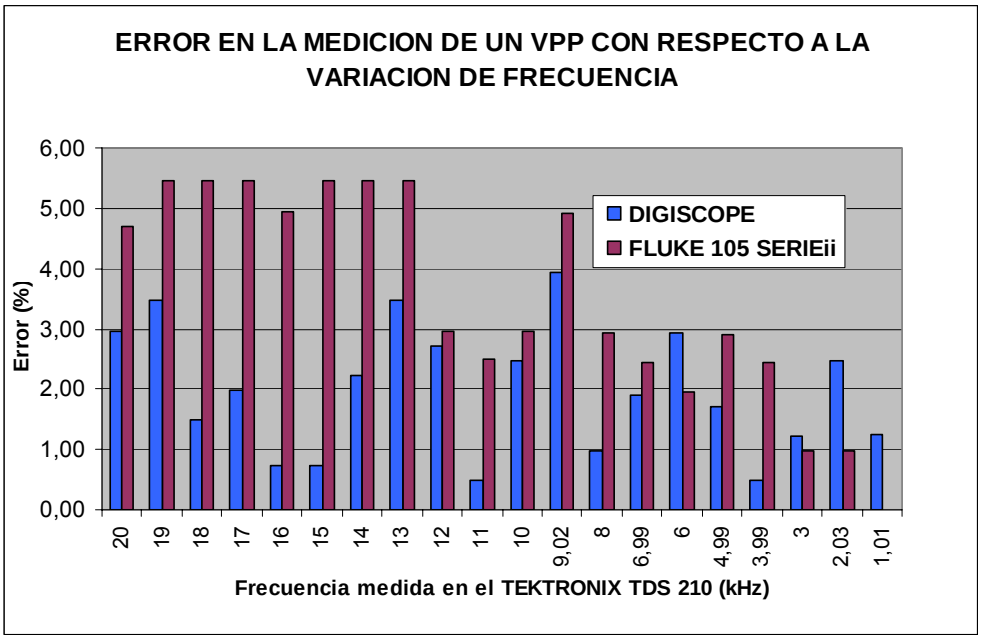
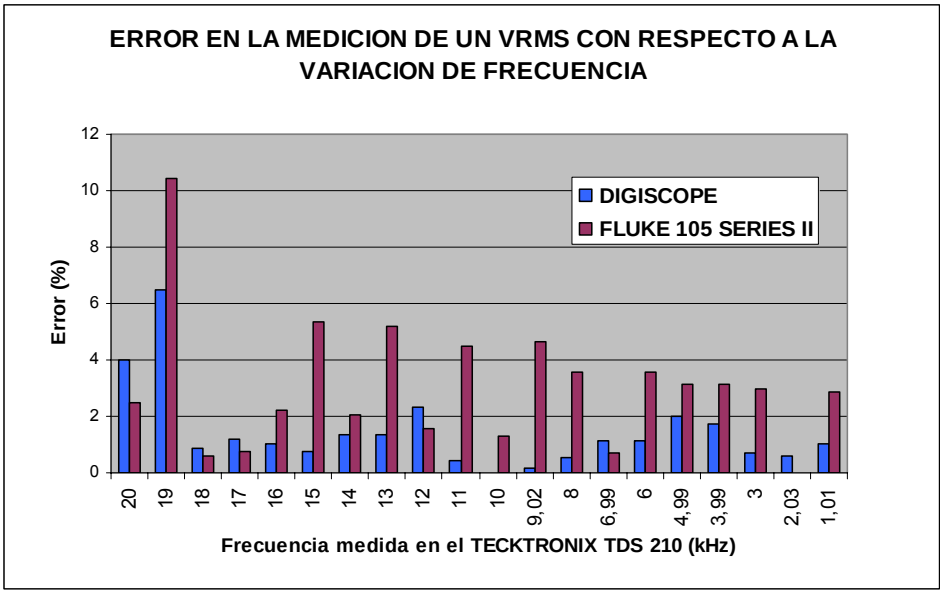


Tabla 17. Porcentajes de error en la medición de valores de Voltaje RMS.

Promedio de la frecuencia medida por TECKTRONIX TDS 210 (Khz.)	Promedio del VRMS medido por TECKTRONIX TDS 210 (V)	ERROR EN DIGISCOPE (%)	ERROR EN FLUKE 105 SERIES II (%)
19,98	0,677	3,988183161	2,511078287
18,995	0,737	6,512890095	10,44776119
17,97	0,676	0,887573964	0,591715976
16,995	0,675	1,185185185	0,740740741
15,985	0,675	1,037037037	2,222222222
15,015	0,676	0,73964497	5,325443787
13,99	0,674	1,335311573	2,077151335

12,975	0,675	1,333333333	5,185185185
11,99	0,691	2,315484805	1,591895803
11,015	0,691	0,434153401	4,486251809
10,005	0,691	0	1,302460203
9,015	0,692	0,144508671	4,624277457
7,995	0,705	0,567375887	3,546099291
6,99	0,705	1,134751773	0,709219858
6	0,705	1,134751773	3,546099291
4,99	0,702	1,994301994	3,133903134
3,99	0,702	1,709401709	3,133903134
3	0,701	0,713266762	2,995720399
2,025	0,7	0,571428571	0
1,0135	0,7	1	2,857142857

Figura 58. Error en la medición de valores de voltaje RMS.



En las mediciones de los parámetros de la señal los más altos errores se presentan en valores de frecuencias altas debido a que en estos rangos el número de muestras por periodo que el conversor Analógico /Digital del DSP alcanza a detectar es relativamente pequeño y por ende el cálculo de estos parámetros puede presentar algunos valores incorrectos.

4.3. PRUEBAS SOBRE LAS MEDICIONES DE PARÁMETROS CON UNA SEÑAL DE FRECUENCIA CONSTANTE.

Como se describió en el capítulo segundo dentro del acondicionamiento de la señal se implementó un sistema de *AUTOSET* (ajuste automático) en el cual el DSP selecciona el nivel de atenuación dependiendo de los voltajes máximo y mínimo de la señal por medio de *switches* analógicos. Para verificar los parámetros de la señal medidos dentro de cada uno de los rangos de atenuación se acopló al prototipo *DIGISCOPE* y a los dos osciloscopios digitales mencionados anteriormente una señal senoidal de frecuencia 10kHz utilizando un generador de señales y se realizaron variaciones en su valor de Voltaje pico a pico, Valor promedio, Valor RMS y frecuencia obteniéndose los siguientes resultados:

Tabla 18. Mediciones de voltaje pico a pico con frecuencia de 10kHz.

NIVEL DE ATENUACION	VOLTAJE PICO A PICO (V)	TEKTRONIX TDS 210 (V)	*DIGISCOPE* (V)	FLUKE 105 SERIES II (V)
1x	1	1 - 1,02	1,04 - 1,08	1
	2	1,98 - 2	1,98 - 2,01	1,92 - 1,96
	2,5	2,48 - 2,50	2,49 - 2,51	2,42 - 2,46
	3	3 - 3,02	3,035 - 3,062	2,92 - 3
2x	4	3,96 - 4,04	3,95 - 3,99	3,88 - 3,92
	5	5 - 5,04	4,92 - 4,96	4,88 - 4,92
	6	5,96 - 6	6,11 - 6,2	6 - 6,08
5x	7	6,96 - 7,04	7,14 - 7,28	7,04 - 7,12
	9	8,96 - 9,04	9,07 - 9,12	8,96 - 9,04
	11	10,9 - 11	10,97 - 11,07	10,9
	13	12,9 - 13	12,91 - 13,02	12,8 - 13
	14	13,9 - 14	13,90 - 14,03	14 - 14,2
10x	15	14,9 - 15	15,14 - 15,33	14,8 - 15
	20	20 - 20,2	20,06 - 20,2	20
	25	24,8 - 25,2	25,08 - 25,22	25 - 25,2

Tabla 19. Mediciones de voltaje promedio con frecuencia de 10kHz,

NIVEL DE ATENUACION	VOLTAJE PICO A PICO (V)	TEKTRONIX TDS 210 (V)	*DIGISCOPE* (V)	FLUKE 105 SERIES II (V)
1x	1	0,00892 - 0,010,6	0,015 - 0,029	-0,008
	2	(-0,0243) – (-0,0228)	0,024 - 0,050	-0,020
	2,5	(-0,0286) – (-0,0242)	0,001 - 0,04	-0,020
	3	(-0,0401) – (-0,0379)	0,035 - 0,049	-0,040
2x	4	(-0,0436) – (-0,0409)	(-0,029) - 0,045	-0,040
	5	(-0,0462) – (-0,0434)	(-0,038) - 0,085	-0,040
	6	(-0,0476) – (-0,0434)	0,074 – 0,201	-0,080
5x	7	(-0,0211) – (-0,0139)	0,056 – 0,160	-0,080
	9	(-0,0312) – (-0,0205)	0,064 – 0,226	-0,080
	11	(-0,0432) – (-0,0325)	(-0,0639) – (-0,142)	-0,080
	13	(-0,0422) – (-0,0327)	0,108 – 0,233	-0,080
	14	(-0,0467) – (-0,0362)	0,111 – 0,392	0 – 0,2
0010x	15	(-0,0657) – (-0,0539)	(-0,069) - 0,246	0
	20	(-0,119) – (-0,107)	0,008 - 0,260	(-0,2) – (-0,4)
	25	(-0,135) – (-0,1199	(-0,209) - 0,446	(-0,2) – (-0,4)

Tabla 20. Mediciones de voltaje RMS con frecuencia de 10kHz.

NIVEL DE ATENUACION	VOLTAJE PICO A PICO (V)	TEKTRONIX TDS 210 (V)	*DIGISCOPE* (V)	FLUKE 105 SERIES II (V)
1x	1	0,348	0,348 – 0,354	0,344
	2	0,680 – 0,684	0,668 – 0,675	0,660
	2,5	0,860 – 0,862	0,849 – 0,860	0,840
	3	1,04	1,011 – 1,02	1
2x	4	1,38	1,33 – 1,35	1,32
	5	1,73	1,67 – 1,70	1,68
	6	2,08	2,029 – 2,05	2
5x	7	2,43 – 2,44	2,38 – 2,41	2,4
	9	3,11 – 3,12	3,03 – 3,08	3,04
	11	3,80 – 3,81	3,71 – 3,75	3,84 – 3,92
	13	4,5 – 4,51	4,38 – 4,47	4,48
	14	4,87 – 4,88	4,80 – 4,85	5
10x	15	5,22	5,26 – 5,3	5,2
	20	6,95 – 6,97	6,97 – 7,034	6,8
	25	8,66 – 8,69	8,65 – 8,86	8,6

Tabla 21. Mediciones de frecuencia con respecto al voltaje medido en el TEKTRONIX.

NIVEL DE ATENUACION	VPP medido en el TEKTRONIX (V)	TEKTRONIX TDS 210 (kHz)	*DIGISCOPE* (kHz)	FLUKE 105 SERIES II (kHz)
1x	1 - 1,02	9,98 – 10,03	9,8 – 10	10
	1,98 - 2	9,98 – 10	10	9,8 – 10
	2,48 - 2,50	9,98 – 10	10	10
	3 - 3,02	9,98 – 10	10	10
2x	3,96 - 4,04	9,97 – 10	10	9,8 – 10
	5 - 5,04	9,99 – 10,02	10	10
	5,96 - 6	9,99 – 10,02	10	10
5x	6,96 - 7,04	9,98 – 10	9,8 – 10	9,8 – 10
	8,96 - 9,04	9,96 – 10	9,8 – 10	9,8 – 10
	10,9 - 11	9,98 – 10	10	9,8 – 10
	12,9 - 13	9,98 – 10	9,8 – 10	9,8 – 10
	13,9 - 14	9,97 – 10	10	9,8 – 10
10x	14,9 - 15	10 – 10,02	10	10
	20 - 20,2	9,99 – 10,02	10	10
	24,8 - 25,2	9,98 – 10	10	10

Con el fin de confrontar los valores obtenidos y calcular los porcentajes de error en las mediciones se eligió de nuevo como referencia los valores mostrados por el osciloscopio digital TEKTRONIX TDS 210 y se compararon con las mediciones visualizadas en los otros dos dispositivos.

Tabla 22. Porcentaje de error en las mediciones de Voltaje pico a pico con frecuencia de 10KHz.

NIVEL DE ATENUACION	Promedio del VPP medido por TECKTRONIX TDS 210 (V)	ERROR EN DIGISCOPE (%)	ERROR EN EL FLUKE 105 SERIES II (%)
1x	1,01	4,95049505	0,99009901
	1,99	0,251256281	2,512562814

	2,49	0,401606426	2,008032129
	3,01	1,279069767	1,661129568
2x	4	0,75	2,5
	5,02	1,593625498	2,390438247
	5,98	2,926421405	1,003344482
5x	7	3	1,142857143
	9	1,055555556	0
	10,95	0,639269406	0,456621005
	12,95	0,115830116	0,386100386
	13,95	0,107526882	1,075268817
10x	14,95	1,906354515	0,334448161
	20,1	0,149253731	0,497512438
	25	0,6	0,4

Figura 59. Error en las mediciones del voltaje pico a pico con frecuencia de 10kHz.

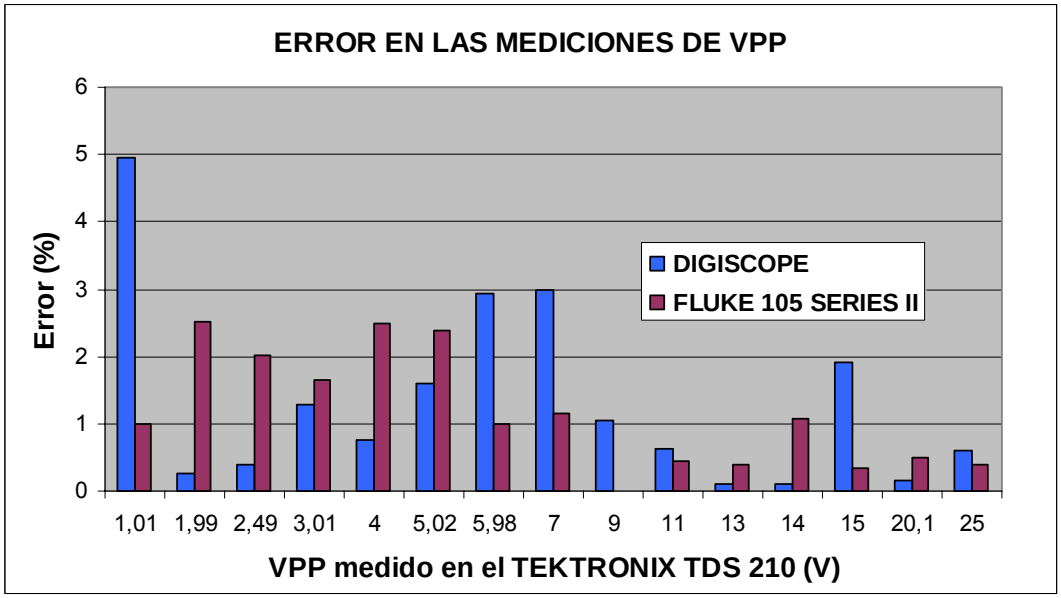


Tabla 23. Porcentaje de error en las mediciones de voltaje RMS con frecuencia de 10kHz.

NIVEL DE ATENUACION	Promedio del VRMS medido por TECKTRONIX TDS 210 (V)	ERROR EN DIGISCOPE (%)	ERROR EN EL FLUKE 105 SERIES II (%)
1x	0,348	0,862068966	1,149425287
	0,682	1,612903226	3,225806452
	0,861	0,81300813	2,43902439
	1,04	2,355769231	3,846153846
2x	1,38	2,898550725	4,347826087
	1,73	2,601156069	2,89017341
	2,08	1,947115385	3,846153846
5x	2,435	1,642710472	1,437371663
	3,115	1,926163724	2,407704655
	3,805	1,97109067	1,97109067
	4,505	1,775804661	0,554938957
	4,875	1,025641026	2,564102564
10x	5,22	1,149425287	0,383141762
	6,96	0,603448276	2,298850575
	8,675	0,922190202	0,864553314

Figura 60. Error en las mediciones del voltaje RMS con frecuencia de 10kHz.

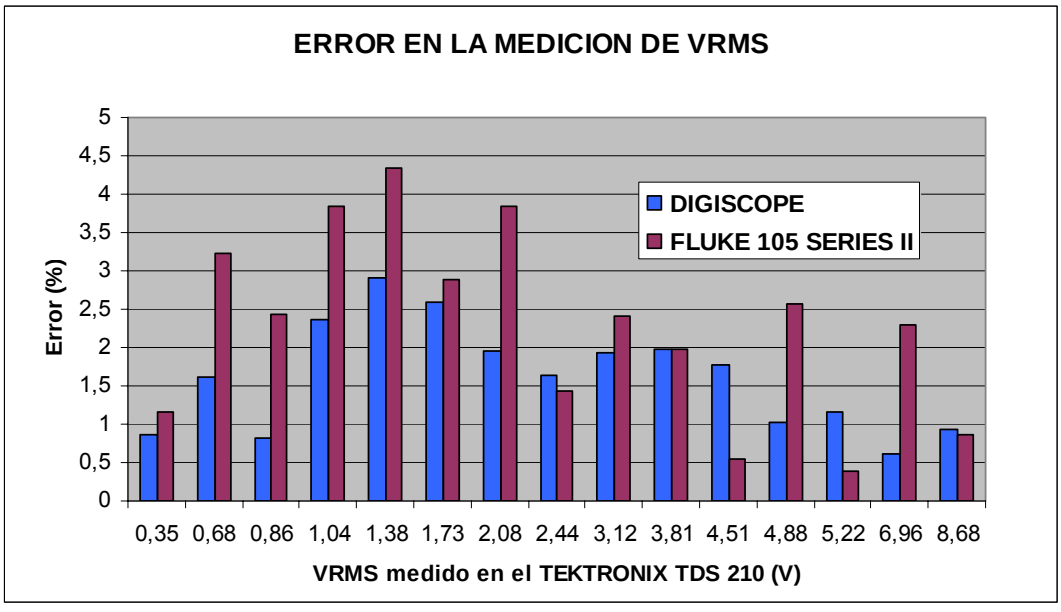
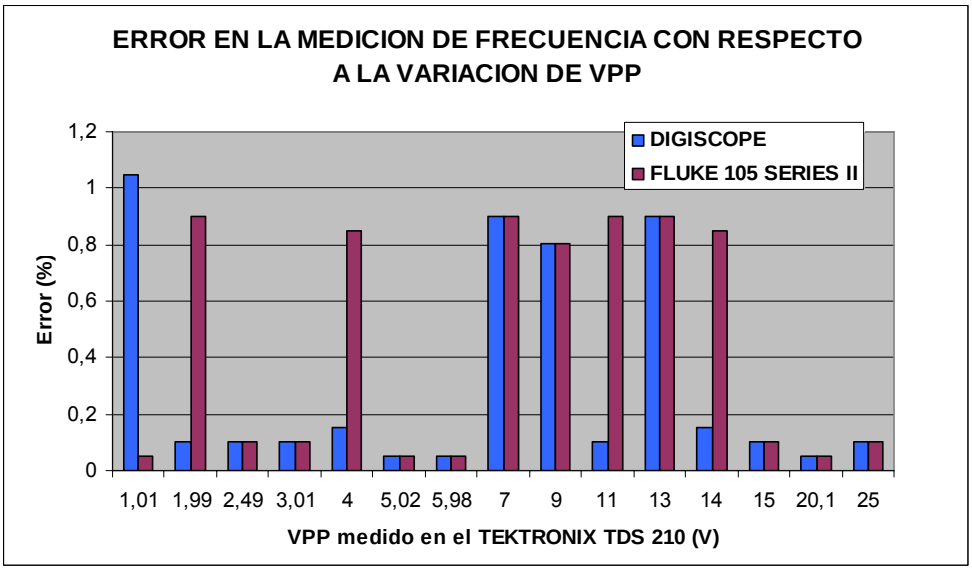


Tabla 24. Porcentaje de error en las mediciones de frecuencia debidas a la variación de voltaje pico a pico.

NIVEL DE ATENUACION	Promedio del VPP medido por TEKTRONIX TDS 210 (V)	Promedio de frecuencia medido por TEKTRONIX TDS 210 (Khz.)	ERROR EN EL DIGISCOPE (%)	ERROR EN EL FLUKE 105 SERIES II (%)
1x	1,01	10,005	1,049475262	0,049975012
	1,99	9,99	0,1001001	0,900900901
	2,49	9,99	0,1001001	0,1001001
	3,01	9,99	0,1001001	0,1001001
2x	4	9,985	0,150225338	0,851276915
	5,02	10,005	0,049975012	0,049975012
	5,98	10,005	0,049975012	0,049975012
5x	7	9,99	0,900900901	0,900900901
	9	9,98	0,801603206	0,801603206
	10,95	9,99	0,1001001	0,900900901
	12,95	9,99	0,900900901	0,900900901
	13,95	9,985	0,150225338	0,851276915
10x	14,95	10,01	0,0999001	0,0999001
	20,1	10,005	0,049975012	0,049975012
	25	9,99	0,1001001	0,1001001

Figura 61. Error en las mediciones de frecuencia debido a la variación de voltaje pico a pico.



Los errores en las mediciones se presentaron, quizás, debido a la inexactitud de los valores de las resistencias utilizadas en el circuito de atenuación (Tolerancia del 5%) y a los grandes intervalos de atenuación que hacen que para señales cuyos voltajes de amplitud sean cercanos a los límites de estos niveles (1.5V, 3V, 7.5V y 15V) la atenuación reduzca significativamente la señal ocasionando errores en la lectura del conversor Analógico/Digital del DSP ya que las muestras son pequeñas en amplitud.

4.3. PRUEBAS SOBRE LAS MEDICIONES DE PARÁMETROS MÁXIMOS Y MÍNIMOS EN UNA SEÑAL DE ENTRADA.

Para obtener los picos máximos y mínimos de voltaje de entrada para el prototipo *DIGISCOPE* se aplicó una señal senoidal de frecuencia 10kHz y se realizó una variación de voltaje pico a pico verificando al mismo tiempo su visualización en la pantalla. El resultado de estas mediciones da como voltaje máximo de señal de entrada 30Vpp y voltaje mínimo de señal de entrada 700mVpp, para voltajes de entrada menores a este valor la señal se presenta muy distorsionada.

El ancho de banda del prototipo está dentro del rango de 30Hz a 20KHz aproximadamente.

4.4. PRUEBAS SOBRE LA VISUALIZACIÓN DE LA FORMA DE ONDA DE UNA SEÑAL DE ENTRADA.

Con el fin de verificar la forma de onda de las señales de entrada se aplicó una señal senoidal, una señal triangular y una señal cuadrada al prototipo *DIGISCOPE* con los siguientes resultados:

Figura 62. Visualización de la forma de onda de una señal senoidal a 60Hz.

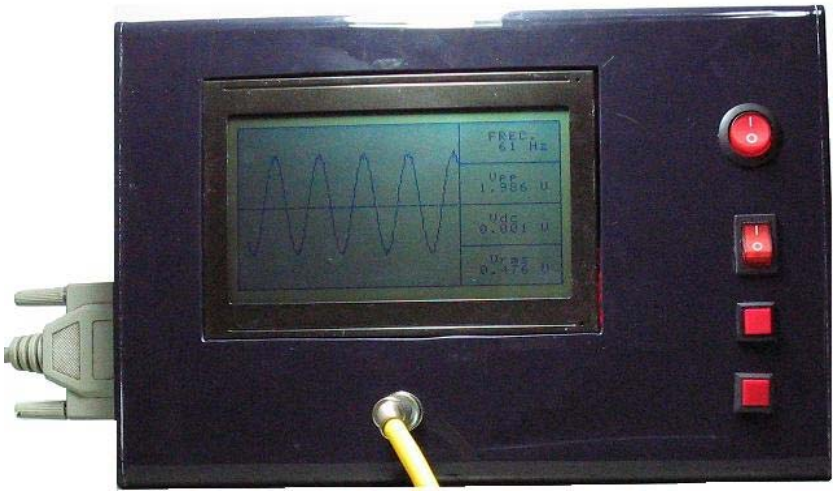


Figura 63. Visualización de la forma de onda de una señal triangular.

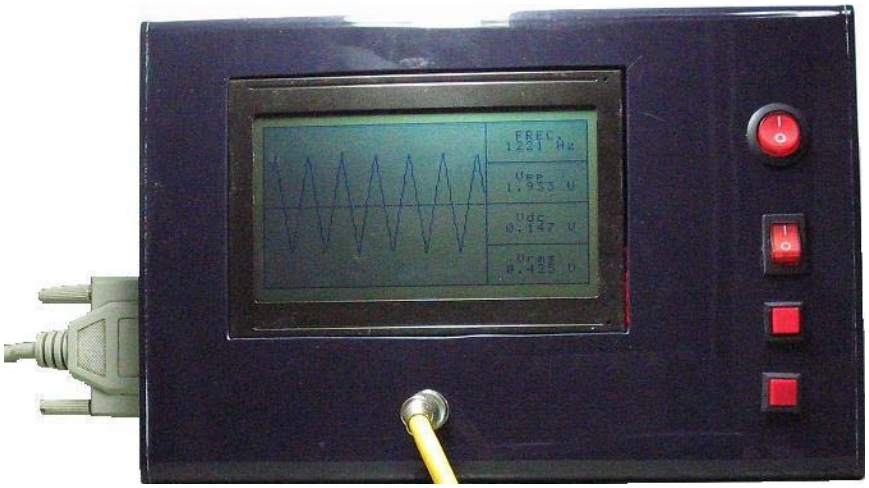
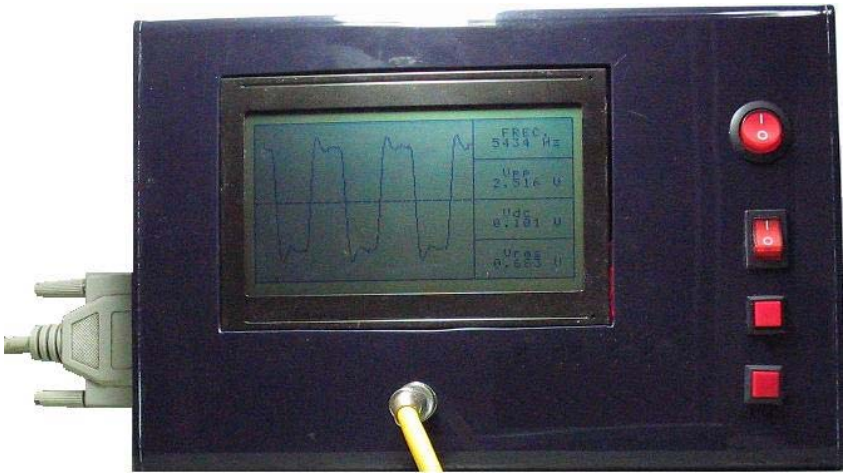


Figura 64. Visualización de la forma de onda de una señal cuadrada a 1,2kHz.

Figura 65. Visualización de la forma de onda de una señal cuadrada a 5,4kHz



La forma de onda de las señales que contienen armónicos tales como la señal cuadrada, presenta distorsión en su visualización debido a que aquellas componentes superiores a 25,6KHz son atenuadas o eliminadas completamente por el filtro análogo utilizado en el prototipo.

5. CONCLUSIONES Y RECOMENDACIONES

- Se construyó una Tarjeta de Desarrollo de propósito general para el DSP 56F8323 de Motorola que permite trabajar con los diferentes periféricos internos del dispositivo como: TIMER, ADC, pines de propósito general, interfase serial y programación.

- Se diseñó e implementó una tarjeta para el prototipo *DIGISCOPE* que permite la programación del DSP, el manejo de la pantalla LCD gráfica y la medición de los parámetros: frecuencia, voltaje pico a pico, valor promedio y valor rms de la señal de entrada. En esta tarjeta sólo se incluyeron los dispositivos necesarios para el funcionamiento específico de esta aplicación.

- El prototipo *DIGISCOPE* y sus posteriores versiones son una alternativa de bajo costo que puede ser útil en los laboratorios de la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. El costo aproximado de los elementos necesarios para el prototipo es de $470.000 pesos.

- Los algoritmos implementados en la medición de los parámetros demostraron ser adecuados para la aplicación ya que en las pruebas realizadas al prototipo se obtuvieron porcentajes de error bajos al comparar las mediciones del prototipo *DIGISCOPE* con las realizadas por los osciloscopios digitales disponibles en los laboratorios de la Escuela (TEKTRONIX TDS 210 y FLUKE 105 SERIES II).

- Los mayores porcentajes de error en las mediciones de los parámetros se presentan debido a los siguientes aspectos: Cuando la frecuencia de la señal de entrada es mayor o igual a 20KHz se obtienen pocas muestras por periodo debido a que la frecuencia máxima de muestreo es de sólo 400KHz, esto ocasiona errores en la medición de los demás parámetros (Vpp, Vdc, Vrms) ya que no se cuenta con suficientes muestras para su cálculo. Cuando la amplitud de la señal de entrada se encuentra alrededor de los límites de los intervalos de atenuación, la amplitud se reduce significativamente y el conversor Analógico / Digital no puede detectar correctamente los valores de las muestras.

- Las señales que contienen armónicos con frecuencias superiores a 25,6KHz se visualizan distorsionadas en la pantalla gráfica debido a que el filtro análogo atenúa estas componentes y en algunos casos las elimina completamente.

- Al realizar la digitalización de la señal se obtienen valores discretos de la misma, por consiguiente para poder mostrar en pantalla una representación continua de la forma de onda es necesario unir estos puntos mediante líneas. Esta fue la estrategia que se utilizó y que permitió obtener una adecuada visualización de las formas de onda. Igualmente, para la visualización de los valores de la medición de los parámetros fue necesario realizar un promedio de ellos antes de enviarlos a la pantalla gráfica, ya que refrescar la pantalla después de cada muestreo dificultaba la visualización e incluso se observaban valores dispersos que por supuesto se corrigieron con el promedio.

- En el acondicionamiento de la señal, aunque la atenuación tiene alta exactitud y precisión, ésta presenta un rango muy limitado y un alto costo. Se recomienda utilizar otra alternativa más económica que permita un

mayor rango en los valores de voltaje de la señal de entrada, como una red de atenuación aunque no discreta sino integrada.

- Para aumentar el ancho de banda del dispositivo es necesario utilizar un Conversor Análogo / Digital externo de alta velocidad que permita obtener suficientes muestras de una señal con componentes de frecuencia relativamente altas, no sólo para el cálculo de sus parámetros, sino para una buena visualización de la forma de onda.

- Para obtener una mejor visualización se recomienda utilizar una pantalla gráfica de mejor resolución ya que los píxeles de la LCD Hyundai HG25504NG-01 son de tamaño 0.47 x 0.47mm. En estos dispositivos se acostumbra utilizar píxeles de tamaño 0.227 x 0.217mm. Además también se recomienda utilizar una LCD con backlight (luz de fondo), característica que también ayuda a mejorar la visualización de las formas de onda y sus parámetros.

- Se recomienda habilitar una estación de trabajo con protecciones eléctricas adecuadas (buena puesta a tierra, libre de estática) que garantice el adecuado funcionamiento y manejo de los integrados según las recomendaciones de los fabricantes.

6. REFERENCIAS BIBLIOGRÁFICAS

[1] FLÓREZ BARRERA, Jairo Iván y HERRERA HERNÁNDEZ, Shirley Paola. Diseño y construcción de un prototipo de un medidor de armónicos de corriente basado en un procesador de señales digitales (DSP). Trabajo de grado (Ingeniero Electrónico). Universidad Industrial de Santander. Facultad Físico Mecánica. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.

[2] OPPENHEIM, Alan y SCHAFER, Ronald. Tratamiento de Señales en Tiempo Discreto. 2 Ed. Prentice Hall, 2000.

[3] KEMMERLY, Jack E , HAYT, William H., y, DURBIN, Steven M. Análisis de Circuitos en Ingeniería. 6 Ed. McGraw Hill, 2003.

7. BIBLIOGRAFÍA

Motorola, Inc. 56F8323 Preliminary Technical Data. MC56F8323/D Revisión 11.0. 2004.

Motorola, Inc. MC56F8300 Peripheral User Manual. MC56F8300UM/D. Revisión 6.0. 2004

Motorola, Inc. 56F8300 Demonstration Board. MC56F8300DBUM/D. Revisión 3.0. 2003

Motorola, Inc. Targeting Motorola 56F8300 Demonstration Board. MC56F8300TUM/D. Revisión 3.0. 2004

PROAKIS, J. G. y MANOLAKIS, D. G. Tratamiento Digital de Señales. 3 Ed. Madrid: Prentice Hall, 1998.

OPPENHEIM, Alan y SCHAFER, Ronald. Tratamiento de Señales en Tiempo Discreto. 2 Ed. Prentice Hall, 2000.

ETTER, Delores M. Solución de problemas de Ingeniería con Matlab. 2 Ed. México. Prentice Hall. 1997

RUSH, Ken y LUCERO-HALL, Terri. Two ways to catch a wave. IEEE Spectrum, 1993.

DELISLE, Norman y GARLAN, David. A Formal Specification of an Oscilloscope. IEEE, 1990.

STUBBINGS, R. M. The Implementation of D.S.P. in Test & Measurement Equipment. IEEE, 1992.

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC. IEEE Standard Pulse Terms and Definitions 194-1977. IEEE, 1977.

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC. IEEE Standard on Pulse Measurement and Analysis by Objective Techniques 181-1977. IEEE, 1977.

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC. IEEE Standard for Terminology and Test Methods for Analog-to-Digital Converters 1241-2000, IEEE, 2001.

FLÓREZ BARRERA, Jairo Iván y HERRERA HERNÁNDEZ, Shirley Paola. Diseño y construcción de un prototipo de un medidor de armónicos de corriente basado en un procesador de señales digitales (DSP). Trabajo de grado (Ingeniero Electrónico). Universidad Industrial de Santander. Facultad Físico Mecánica. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.

AMARIS, Jean Pierre y LOPEZ, José Alberto. Elaboración del software para la caracterización de una celda electromagnética utilizando DSP familia 56800 de Motorota. Bucaramanga, 2003. Trabajo de grado (Ingeniero Electrónico). Universidad Industrial de Santander. Facultad Físico Mecánica. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones.

ANEXO A. CODIGO GENERAL DEL PROGRAMA.

```
#include "Cpu.h"
#include "Events.h"
#include "AD1.h"
#include "FC161.h"
#include "IFsh1.h"
#include "RES.h"
#include "WR.h"
#include "A0.h"
#include "S1.h"
#include "S2.h"
#include "S3.h"
#include "S4.h"
#include "Spaso.h"
#include "Datos.h"
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"
#include "math.h"

#define Derecha 0x4C
#define Izquierda 0x4D
#define Arriba 0x4E
#define Abajo 0x4F
#define TOLER 0.0001
```

```
const byte UIS[]=
      { 0x00,0x01,0xCF,0xF3,0x00,0x00,0x00,0x07,0xCF,0xF3,0xE0,0x00
       ,0x00,0x1F,0xCF,0xF3,0xF8,0x00,0x00,0x1F,0xCF,0xF3,0xFE,0x00
       ,0x01,0x9F,0xCF,0xF3,0xFF,0x80,0x03,0x9F,0xCF,0xF3,0xFF,0xC0
       ,0x07,0x9F,0xCF,0xF3,0xFF,0xE0,0x0F,0x9F,0xCF,0xF3,0xFF,0xF0
       ,0x1F,0x9F,0xCF,0xF3,0xFF,0xF8,0x3F,0x9F,0xCF,0xF3,0xFF,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xFF,0xFC,0x3F,0x9F,0xCF,0xF3,0xFF,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xF8,0x00,0x3F,0x9F,0xCF,0xF3,0xF8,0x00
       ,0x3F,0x9F,0xCF,0xF3,0xFF,0xFC,0x3F,0x9F,0xCF,0xF3,0xFF,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xFF,0xFC,0x3F,0x9F,0xCF,0xF3,0xFF,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xFF,0xFC,0x3F,0x9F,0xCF,0xF3,0xFF,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xFF,0xFC,0x3F,0x9F,0xCF,0xF3,0xFF,0xFC
       ,0x3F,0x9F,0xCF,0xF0,0x01,0xFC,0x3F,0x9F,0xCF,0xF0,0x01,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC
       ,0x3F,0x9F,0xCF,0xF3,0xF9,0xFC,0x3F,0x9F,0xCF,0xF3,0xFF,0xFC
       ,0x3F,0xFF,0xCF,0xF3,0xFF,0xFC,0x3F,0xFF,0xCF,0xF3,0xFF,0xFC
       ,0x3F,0xFF,0xCF,0xF3,0xFF,0xFC,0x3F,0xFF,0xCF,0xF3,0xFF,0xFC
       ,0x1F,0xFF,0xCF,0xF3,0xFF,0xF8,0x0F,0xFF,0xCF,0xF3,0xFF,0xF0
       ,0x07,0xFF,0xCF,0xF3,0xFF,0xE0,0x03,0xFF,0xCF,0xF3,0xFF,0xC0
       ,0x01,0xFF,0xCF,0xF3,0xFF,0x80,0x00,0xFF,0xCF,0xF3,0xFF,0x00
       ,0x00,0x3F,0xCF,0xF3,0xFC,0x00,0x00,0x1F,0xCF,0xF3,0xF8,0x00
```

```
        ,0x00,0x07,0xCF,0xF3,0xE0,0x00,0x00,0x00,0xCF,0xF3,0x00,0x00
        ,0x00,0x00,0x07,0xE0,0x00,0x00};

const byte mascara[]=
{
         128,64,32,16,8,4,2,1
};

const byte FRE[]={"FREC."};
const byte VPP[]={"Vpp"};
const byte VRMS[]={"Vrms"};
const byte VDC[]={"Vdc"};
const byte HZ[]={"Hz"};
const byte VOLT[]={"V"};
const byte UNIV[]={"UNIVERSIDAD INDUSTRIAL"};
const byte STDER[]={"DE SANTANDER"};
const byte E3T[]={"E3T"};
const byte CEMOS[]={"GRUPO CEMOS"};
const byte OSC[]={"DIGISCOPE - 2005"};
const byte JULIAN[]={"JULIAN EDUARDO BADILLO H."};
const byte CATE[]={"ANYELA CATERINE CABRA B."};

void ESCRIBIR_COMANDO(byte comando);
void ESCRIBIR_DATO(byte dato);
void mostrarV(short valor[], int posicion, byte longitud, byte direccion);
void mostrar(byte valor, int posicion, int longitud, byte direccion);
void mostrarT(const byte texto[],int posicion, byte longitud,byte direccion);
void CSR_DIR(byte direccion);
void CSR_POS(word posicion);
void INIC_LCD(void);
```

```
void CLR_TEXTO(void);
void CLR_GRAFICO(void);
void PORTADA(void);
void MARCOS(void);
void TEXTO(void);
void MUESTREO(void);
void CODIGO(float variable);
float sqrt(float x);

int s[336];
short digV[7],cod[7],codVdc[7],codVpp[7],codrms[7];
int a;
void main(void)
{float
max,min,caso,Vdc1[10],Vpp1[10],rms1[10],h,cuadrado[336],y[336],Vdc,Vpp,r
ms,muestra[336];
 word periodo;
 byte n,t,filas[168],r,p,v,control,signo,ref;
 int j,pos,rep[168],f;
 short datos[128],m,digf[5],codf[5];

  Vdc=0; // Voltaje Promedio
  n=0; // Muestras por Periodo
  h=0;  // Tamaño de Paso
  control=0; // Variable de control de visualizacion

  INIC_LCD(); // INICIALIZACION LCD

  PORTADA(); // PORTADA
  CLR_TEXTO(); // LIMPIAR CAPA DE TEXTO
```

```
CLR_GRAFICO(); // LIMPIAR CAPA DE GRAFICOS
MARCOS(); // MARCOS
TEXTO(); // IMPRIME TEXTO DE PARAMETROS

for(j=0;j<10;j++)
{
	Vdc1[j]=0;
	Vpp1[j]=0;
	rms1[j]=0;
}

AD1_Measure(FALSE); // Bean Measure

S1_SetVal(); // Nivel de Atenuación 1x
Spaso_ClrVal(); // Habilitar Capacitor de Paso
setReg(ADCA_ADCR2,5); // Selección de Divisor de Reloj
setReg(ADCA_ADCR1,4);		// Modo Triggered Sequential
setRegBit(ADCA_ADCR1,ZCIE); // Habilitar Cruces por Cero
clrRegBit(ADCA_ADCR1,STOP);

for(j=0;j<30000;j++)
{
	setRegBit(ADCA_ADCR1,START);
	while(getRegBit(ADCA_ADSTAT,CIP)){} // Conversion in Progress
	for(a=0;a<100;a++){}
	if(!getRegBit(ADCA_ADCR1,ZCIE))
		break;
}

for(;;)
```

```
{
    if(getRegBit(ADCA_ADCR1,ZCIE)==0)

        IFsh1_GetPage(8192); // Get virtual page from FLASH
        IFsh1_GetWordPage(0,&periodo); // Get word from address in
        virtual page

        f=1/(periodo*0.000001);
        if(f<3000)
        {
            if(f<1500)
            {
                setReg(ADCA_ADCR2,5); // Selección de Divisor
                de Reloj
                setReg(ADCA_ADCR1,4); // Modo Triggered
                Sequential
                setRegBit(ADCA_ADCR1,ZCIE); // Habilitar
                Cruces por Cero
                clrRegBit(ADCA_ADCR1,STOP);

                for(j=0;j<30000;j++)
                {
                    setRegBit(ADCA_ADCR1,START);
                    while(getRegBit(ADCA_ADSTAT,CIP)){} //
                    Conversion in Progress
                    for(a=0;a<400;a++){}
                    if(!getRegBit(ADCA_ADCR1,ZCIE))
                        break;
                }
```

```
                IFsh1_GetPage(8192); // Get virtual page from
                FLASH
                IFsh1_GetWordPage(0,&periodo); // Get word from
                address in virtual page

                if(periodo>0.000666)
                        f=1/(periodo*0.000001);
        }
        else
        {
                setReg(ADCA_ADCR2,20); // Selección de Divisor
                de Reloj
                setReg(ADCA_ADCR1,2);// Modo Loop Sequential
                setRegBit(ADCA_ADCR1,ZCIE); // Habilitar
                Cruces por Cero
                clrRegBit(ADCA_ADCR1,STOP);
                setRegBit(ADCA_ADCR1,START);

                while(getRegBit(ADCA_ADCR1,ZCIE)){}
                IFsh1_GetPage(8192); // Get virtual page from
                FLASH
                IFsh1_GetWordPage(0,&periodo); // Get word from
                address in virtual page

                f=1/(periodo*0.000001);
        }
}
else
{
```

```
        setReg(ADCA_ADCR2,5); // Selección de Divisor de
        Reloj
        setReg(ADCA_ADCR1,2);// Modo Loop Sequential
        setRegBit(ADCA_ADCR1,ZCIE); // Habilitar Cruces por
        Cero
        clrRegBit(ADCA_ADCR1,STOP);
        setRegBit(ADCA_ADCR1,START);

        while(getRegBit(ADCA_ADCR1,ZCIE)){}
        IFsh1_GetPage(8192); // Get virtual page from FLASH
        IFsh1_GetWordPage(0,&periodo); // Get word from
        address in virtual page

        f=1/(periodo*0.000001);
}

/* DETERMINAR n y h */
n=400000/f; // Muestras por Periodo
j=5; // Variable para selección de divisor de reloj
h=0.0000025; // Tamaño de Paso
a=0;
if(f<5000)
{
        if(f<1500)
        {
                j=29;
                n=38699/f;
                h=0.00002584;
                a=40; // Retardo para Visualizacion
                if(f<500)
```

```
                {
                        j=29;
                        n=11956/f;
                        h=0.00008364;
                        a=170; // Retardo para Visualizacion
                        if(f<200)
                        {
                                j=29;
                                n=3708/f;
                                h=0.00026962;
                                a=600; // Retardo para Visualizacion
                                if(f<100)
                                {
                                        j=29;
                                        n=2262/f;
                                        h=0.000441915;
                                        a=1000; // Retardo para
                                        Visualizacion
                                }
                        }
                }
        }
        else
        {
                j=15;
                n=71736/f;
                h=0.00001394;
                a=10;
        }
}
```

```
/* AJUSTE HORIZONTAL */
setReg(ADCA_ADCR2,j); // Selección de Divisor de Reloj
S1_SetVal(); // Nivel de Atenuación 1x
//S2_ClrVal(); // Nivel de Atenuación 2x

/* MUESTREO PRELIMINAR */
MUESTREO();
for(j=0;j<336;j++)
{
        y[j]=s[j];
}

/* CÁLCULO DEL PICO POSITIVO Y EL PICO NEGATIVO */
max=y[0];
min=y[0];

for(j=1;j<336;j++)
{
        if(y[j]>max)
                max=y[j];
        if(y[j]<min)
                min=y[j];
}

max=(max*1.48)/14888.0;
min=(min*1.48)/14888.0;

max=fabsf(max); // Valor Absoluto del Máximo
min=fabsf(min); // Valor Absoluto del Mínimo
```

```
if(max>=min)
        caso=max;
else
        caso=min;

/* SELECCIÓN DEL NIVEL DE ATENUACIÓN */
if(caso>7.5)
{
        for(j=0;j<336;j++)
        {
                y[j]=(s[j]*14.5)/14888.0; // VECTOR DE
                MUESTRAS EN VOLTS
        }
        m=4;//Calculo Filas
}
else
{
        if(caso>3)
                if(caso<=7.5)
                {
                        S3_SetVal(); // NIVEL DE ATENUACIÓN 5X
                        S4_ClrVal(); // NIVEL DE ATENUACIÓN
                        10X
                        MUESTREO();
                        for(j=0;j<336;j++)
                        {
                                y[j]=(s[j]*7.12)/14888.0;  // VECTOR
                                DE MUESTRAS EN VOLTS
                        }
```

```
                m=8;//Calculo Filas
            }

        if(caso>1.5)
            if(caso<=3)
            {
                S2_SetVal(); // NIVEL DE ATENUACIÓN 2X
                S4_ClrVal(); // NIVEL DE ATENUACIÓN
                10X
                MUESTREO();
                for(j=0;j<336;j++)
                {
                    y[j]=(s[j]*2.85)/14888.0;  // VECTOR
                    DE MUESTRAS EN VOLTS
                }
                m=20;//Calculo Filas
            }
        if(caso<=1.5)
        {
            S1_SetVal(); // NIVEL DE ATENUACIÓN 1X
            S4_ClrVal(); // NIVEL DE ATENUACIÓN 10X
            MUESTREO();
            for(j=0;j<336;j++)
            {
                y[j]=(s[j]*1.48)/14888.0;  // VECTOR DE
                MUESTRAS EN VOLTS
            }
            m=40;//Calculo Filas
        }
}
```

```
a=0; // Reinicializar el Retardo de muestras

/* CALCULO PARAMETROS */
max=y[0];
min=y[0];
n=2*n; // Numero de muestras para dos periodos

for(j=0;j<n;j++)
{
        Vdc1[control]=Vdc1[control]+y[j];
        cuadrado[j]=(y[j]*y[j]);
        if(y[j]>max)
                max=y[j];
        if(y[j]<min)
                min=y[j];
}
Vdc1[control]=(Vdc1[control]/n); // Voltaje promedio
Vpp1[control]=max-min; // Voltaje Pico a Pico

// Cálculo del Valor RMS
t=n%2;

if(t>0)
{
        for(j=0;j<(n-1);j=j+2)
        {

rms1[control]=rms1[control]+2*(cuadrado[j]+4*cuadrado[j+1]+cuadrado
[j+2])/6; // Simpson 1/3
        }
```

```
        }
        else
        {

rms1[control]=rms1[control]+3*(cuadrado[0]+3*(cuadrado[1]+cuadrado[
2])+cuadrado[3])/8; // Simpson 3/8

                for(j=3;j<(n-1);j=j+2)
                {
rms1[control]=rms1[control]+2*(cuadrado[j]+4*cuadrado[j+1]+cuadrado
j+2])/6; // Simpson 1/3
                }
        }
        rms1[control]=h*rms1[control]*f/2;
        rms1[control]=sqrt(rms1[control]);  // Valor RMS
        control=control+1;

        /* CODIGO ASCII PARAMETROS */
        if(control==10)
        {
                Vdc=0;
                Vpp=0;
                rms=0;

                for(j=0;j<10;j++)
                {
                        Vdc=Vdc+Vdc1[j];
                        Vpp=Vpp+Vpp1[j];
                        rms=rms+rms1[j];
                }
```

```
Vdc=Vdc/10; // Promedio de las mediciones de
parámetros
Vpp=Vpp/10;
rms=rms/10;

// Codigo ASCII Vdc
CODIGO(Vdc);
for(j=0;j<7;j++)
{
        codVdc[j]=cod[j];
}

// Codigo ASCII Vpp
CODIGO(Vpp);
for(j=0;j<7;j++)
{
        codVpp[j]=cod[j];
}

// Codigo ASCII rms
CODIGO(rms);
for(j=0;j<7;j++)
{
        codrms[j]=cod[j];
}

mostrarV(codVpp,0xDC,7,Izquierda); // VALOR Vpp
mostrarV(codVdc,0x15C,7,Izquierda); // VALOR Vdc
mostrarV(codrms,0x1DC,6,Izquierda); // VALOR rms
control=0;
```

```
        for(j=0;j<10;j++)
        {
                Vdc1[j]=0;
                Vpp1[j]=0;
                rms1[j]=0;
        }

}

/* AJUSTE VISUALIZACIÓN */
for(j=0;j<336;j++)
{
        muestra[j]=y[j]-Vdc;
}

for(j=0;j<335;j++)
{
        if(muestra[j]<0)
        {
                signo=muestra[j]*muestra[j+1];
                if(signo<0)
                        ref=j;
        }
}

/* IMPRIMIR DATOS LCD */

// CALCULO FILAS
for(j=0;j<168;j++)
```

```
{
        filas[j]=((y[j+ref]*m+60)-120)*(-1); // Cálculo de la Fila
}

ref=0; // Reinicializar variable de referencia de Visualizacion

for(j=0;j<167;j++)
{
        rep[j]=filas[j+1]-filas[j];  // Interpolacion Grafica
}
pos=0x1081;
m=0;
r=0;
p=0;
v=0;
for (t=0;t<21;t++)
{
        for(j=0;j<128;j++)
        {
                datos[j]=0; // Inicialización vector de datos LCD
        }

        for(j=0;j<8;j++)
        {
                r=fabsf(rep[v]);
                p=filas[m];
                m=m+1;
                for(n=0;n<r;n++)
                {
```

```
                    datos[p]=datos[p]+mascara[j]; // Elementos
                    del vector datos LCD
                    if(rep[v]>0)
                        p=p+1;
                    if(rep[v]<0)
                        p=p-1;
                }
                if(r==0)
                {
                    datos[p]=datos[p]+mascara[j]; // Elementos
                    del vector datos LCD
                }
                v=v+1;
        }
        mostrarV(datos,pos,121,Abajo); // Imprimir Grafica LCD
        pos=pos+0x01;
}

// Codigo ASCII Frecuencia
for(j=0;j<5;j++)
{
        codf[j]=0x20;
}
for(j=0;j<5;j++)
{
        digf[j]=f%10; // Vector de dígitos
        f=f/10;
        codf[j]=0x30+digf[j]; // Vector de código ASCII
        if(f==0)
            break;
```

```
}

/* IMPRIME VALORES DE PARAMETROS */
mostrar(0xFE,0x1800,22,Derecha);//LINEA 0 VOLTS
mostrarV(codf,0x5B,5,Izquierda); // VALOR FRECUENCIA

setReg(ADCA_ADCR2,5); // Selección de Divisor de Reloj
setReg(ADCA_ADCR1,4);// Modo Triggered Sequential
setRegBit(ADCA_ADCR1,ZCIE); // Habilitar Cruces por Cero
Spaso_ClrVal(); // Habilitar Capacitor de Paso
S1_SetVal(); // Nivel de Atenuación 1x
S4_ClrVal(); // NIVEL DE ATENUACIÓN 10X
clrRegBit(ADCA_ADCR1,STOP);

/* MEDICION DE FRECUENCIA */

for(j=0;j<30000;j++)
{
        setRegBit(ADCA_ADCR1,START);
        while(getRegBit(ADCA_ADSTAT,CIP)){} // Conversion in
        Progress
        for(a=0;a<100;a++){}
        if(!getRegBit(ADCA_ADCR1,ZCIE))
                break;
}

for(n=0;n<10;n++)
{
        for(j=0;j<32000;j++){}
}
```

```
        }
  }
}

/* FUNCIONES */
void MUESTREO(void)
{int i,b;

        Spaso_SetVal(); // Deshabilitar Capacitor de Paso
        setReg(ADCA_ADCR1,4);// Modo Triggered Sequential
        clrRegBit(ADCA_ADCR1,STOP);

        for(i=0;i<300;i++){}
        for(i=0;i<336;i++)
        {
                setRegBit(ADCA_ADCR1,START); // Inicio de conversión
                while(getRegBit(ADCA_ADSTAT,CIP)){} // Conversion in
                Progress
                s[i]=getReg(ADCA_ADRSLT0); // Vector de Muestras
                for(b=0;b<a;b++){}
        }
        S4_SetVal(); // Habilitar NIVEL DE ATENUACIÓN 10X
        S3_ClrVal(); // Deshabilitar NIVEL DE ATENUACIÓN 5X
        S2_ClrVal(); // Deshabilitar NIVEL DE ATENUACIÓN 2X
        S1_ClrVal(); // Deshabilitar NIVEL DE ATENUACIÓN 1X
        Spaso_ClrVal(); // Habilitar Capacitor de Paso
 }

void CODIGO(float variable)
{int vol;
```

```
byte i;
	vol=variable*1000;
	for(i=0;i<7;i++)
	{
		digV[i]=0x20; // Limpiar con espacios
		cod[i]=0x30; // Limpiar con ceros
	}
	cod[3]=0x2E; // Codigo ASCII "."
	cod[5]=0x20; // Codigo ASCII ""

	if(vol<0)
	{
		vol=vol*(-1);
		cod[6]=0x2D; // Código ASCII "-"
	}
	else
		cod[6]=0x20;

	//Digitos decimales
	for(i=0;i<3;i++)
	{
		digV[i]=vol%10; // Vector de digitos
		vol=vol/10;
		cod[i]=0x30+digV[i]; // Vector de código ASCII
		if(vol==0)
			break;
	}
	//Digitos enteros
	if(vol==0)
	{
```

```
            cod[4]=0x30; // Código ASCII "0"
        }
        else
        {
            for(i=4;i<6;i++)
            {
                digV[i]=vol%10; // Vector de dígitos
                vol=vol/10;
                cod[i]=0x30+digV[i]; // Vector de código ASCII
                if(vol==0)
                    break;
            }
        }
}

void ESCRIBIR_DATO(byte dato)
{int i;
  Datos_PutVal(dato);
  for(i=0;i<300;i++){}
  WR_ClrVal(); // WR
  for(i=0;i<300;i++){}
  WR_SetVal(); // WR
}

void ESCRIBIR_COMANDO(byte comando)
{int i;
  A0_SetVal(); // AO
  Datos_PutVal(comando); // C
  WR_ClrVal(); // WR
  for(i=0;i<300;i++){}
```

```
  WR_SetVal(); // WR
  A0_ClrVal(); // AO
}

/*MUESTRA EN PANTALLA EL VALOR COMO VECTOR*/
void mostrarV(short valor[], int posicion, byte longitud, byte direccion)
{
        int i;
        CSR_POS(posicion);
        CSR_DIR(direccion);
        ESCRIBIR_COMANDO(0x42); // MWRITE
        for(i=0;i<=longitud-1;i++)
        {
                ESCRIBIR_DATO(valor[i]);
        }
}

//MUESTRA EN PANTALLA EL VALOR//
void mostrar(byte valor, int posicion, int longitud, byte direccion)
{
        int i;
        CSR_POS(posicion);
        CSR_DIR(direccion);
        ESCRIBIR_COMANDO(0x42); // MWRITE
        for(i=0;i<=longitud-1;i++)
        {
                ESCRIBIR_DATO(valor);
        }
}
```

```
//MUESTRA TEXTO//
void mostrarT(const byte texto[],int posicion, byte longitud,byte direccion)
{
        int i;
        CSR_POS(posicion);
        CSR_DIR(direccion);
        ESCRIBIR_COMANDO(0x42);
        for(i=0;i<=longitud-1;i++)
        ESCRIBIR_DATO(texto[i]);
}

/*DIRECCION DE MOVIMIENTO DEL CURSOR*/
void CSR_DIR(byte direccion)
{
        ESCRIBIR_COMANDO(direccion);
}

/*POSICION DEL CURSOR*/
void CSR_POS(word posicion)
{
        ESCRIBIR_COMANDO(0x46); // CSRW Posicion del Cursor
        ESCRIBIR_DATO(posicion);
        ESCRIBIR_DATO(posicion>>8);
}

/* INICIALIZACION LCD*/
void INIC_LCD(void)
{
  WR_SetVal(); // WR
  // COMANDO
```

```
ESCRIBIR_COMANDO(0x40); // SYSTEM SET
// DATOS
ESCRIBIR_DATO(0x30); // P1
ESCRIBIR_DATO(0x87); // P2
ESCRIBIR_DATO(0x07); // P3
ESCRIBIR_DATO(0x4F); // P4
ESCRIBIR_DATO(0x52); // P5
ESCRIBIR_DATO(0x7F); // P6
ESCRIBIR_DATO(0x20); // P7
ESCRIBIR_DATO(0x00); // P8
// COMANDO
ESCRIBIR_COMANDO(0x44); // SCROLL
// DATOS PANTALLA 1
ESCRIBIR_DATO(0x00); // P1
ESCRIBIR_DATO(0x00); // P2
ESCRIBIR_DATO(0x7F); // P3
// DATOS PANTALLA 2
ESCRIBIR_DATO(0x00); // P4
ESCRIBIR_DATO(0x10); // P5
ESCRIBIR_DATO(0x7F); // P6
// DATOS PANTALLA 3
ESCRIBIR_DATO(0x00); // P7
ESCRIBIR_DATO(0x20); // P8
// COMANDO
ESCRIBIR_COMANDO(0x5A); // HDOT SCR
ESCRIBIR_DATO(0x00); // P1
// COMANDO
ESCRIBIR_COMANDO(0x5B); // OVLAY
ESCRIBIR_DATO(0x01); // P1
// LIMPIAR TEXTO
```

```
  CLR_TEXTO();
  // LIMPIAR GRAFICO
  CLR_GRAFICO();
  // COMANDO
  ESCRIBIR_COMANDO(0x5D); // CSRFORM
  ESCRIBIR_DATO(0x00); // P1
  ESCRIBIR_DATO(0x81); // P2
  // COMANDO
  ESCRIBIR_COMANDO(0x59); // DIPS ON
  ESCRIBIR_DATO(0x56); // P1
  //FIN INICIALIZACION
}

void CLR_TEXTO(void)
{
  mostrar(0x20,0x00,512,Derecha);//Arriba
}

void CLR_GRAFICO(void)
{
  mostrar(0x00,0x1000,4096,Derecha);//Arriba
}

void PORTADA(void)
{byte k,m;
 int j;
 word Direccion;

  // LOGO UIS
  j=0;
```

```
  Direccion=0x100D;
  ESCRIBIR_COMANDO(0x4C); // CSRDIR Direccion del Movimiento del
Cursor
  ESCRIBIR_COMANDO(0x42); // MWRITE
  for(k=1;k<=59;k++)
  {
        ESCRIBIR_COMANDO(0x46); // CSRW Posición del Cursor
        ESCRIBIR_DATO(Direccion); // P1 Posicion L
        ESCRIBIR_DATO(Direccion>>8); // P1 Posicion H
        ESCRIBIR_COMANDO(0x42); // MWRITE Comando para escribir

        for(m=1;m<=6;m++)
        {
              ESCRIBIR_DATO(UIS[j]);
              j++;
        }
        Direccion=Direccion+0x20;
  }
  mostrarT(UNIV,0x105,22,Derecha); // TEXTO UNIV
  mostrarT(STDER,0x129,12,Derecha); // TEXTO STDER
  mostrarT(E3T,0x16E,3,Derecha); // TEXTO E3T
  mostrarT(CEMOS,0x18A,11,Derecha); // TEXTO CEMOS
  mostrarT(OSC,0x1A8,16,Derecha); // TEXTO OSC
  mostrarT(JULIAN,0x1C4,25,Derecha); // TEXTO JULIAN
  mostrarT(CATE,0x1E4,24,Derecha); // TEXTO CATE
}

void MARCOS(void)
{
  CLR_GRAFICO();
```

```
  mostrar(0xFF,0x1000,32,Derecha);//Arriba
  mostrar(0x80,0x1020,126,Abajo);//Izquierda
  mostrar(0x1,0x103F,126,Abajo);//Derecha
  mostrar(0xFF,0x1FE0,32,Derecha);//Abajo
  mostrar(0x80,0x1036,126,Abajo);//DIVISION
  mostrar(0xFF,0x1416,10,Derecha);//LINEA 1
  mostrar(0xFF,0x1816,10,Derecha);//LINEA 2
  mostrar(0xFF,0x1C16,10,Derecha);//LINEA 3
}
void TEXTO(void)
{
  mostrarT(FRE,0x39,5,Derecha); // TEXTO FRECUENCIA
  mostrarT(HZ,0x5D,2,Derecha); // TEXTO HZ
  mostrarT(VPP,0xB9,3,Derecha); // TEXTO Vpp
  mostrarT(VOLT,0xDE,1,Derecha); // TEXTO V
  mostrarT(VDC,0x139,3,Derecha); // TEXTO Vdc
  mostrarT(VOLT,0x15E,1,Derecha); // TEXTO V
  mostrarT(VRMS,0x1B9,4,Derecha); // TEXTO Vrms
  mostrarT(VOLT,0x1DE,1,Derecha); // TEXTO V
}
float sqrt(float x)
{
 if(x==0) return 0;
  else
   {
    if(x==1) return 1;
     else
      {
       float t=x/2,j=2*TOLER;
       while(j>=TOLER)
```

```
      {
      t=(t+x/t)/2;
      if((t*t-x)*100/x>=0)
        {j=(t*t-x)*100/x;}
      else
        {j=(x-t*t)*100/x;}
      }
      return t;
    }
  }
}

////////////////////////////////////// INTERRUPCION ////////////////////////////////////////

void AD1_OnZeroCrossing(void)
{word time,a[15],may=0;
 k=k+1;

 if(k==1)
        FC161_Reset(); // reset the counter
 else
 {
        if(FC161_GetTimeUS(&time)==ERR_OK)
                a[z]=time;
        k=0;
        z++;
        if (z==15)
        {
                clrRegBit(ADCA_ADCR1,ZCIE); // Deshabilitacion de Cruces
                por Cero
```

```
                for(z=1;z<15;z++)
                {
                        if(a[z]>may)
                                may=a[z];
                }
                IFsh1_GetPage(8192); // Get virtual page from FLASH
                IFsh1_SetWordPage(0,may); // Write word to address in virtual
                page
                IFsh1_SetPage(8192); // Write virtual page to address in FLASH
        z=0;
   }
   FC161_Reset(); // reset the counter
 }
}
```

ANEXO B. PROGRAMA DEL FILTRO DIGITAL EN MATLAB.

```
clear
clc
orden=input('Introduzca el orden del filtro: ');
fc=input('Introduzca la frecuencia de corte: ');

close all
w=2*pi*fc;
wn=w/(pi*fs);
wn1=w/(pi*fs1);

[b]=fir1(orden,wn,kaiser((orden+1),9.84086));
[H] = freqz(b,1,512);
fas=phasez(b,1,length(H),fs);
W=linspace(0,fs/2,length(H));

subplot(2,1,1),plot(W,20*log10(abs(H))),
title('Filtro con frecuencia de muestreo 400KHz'),
xlabel('Frecuencia en Hz'),
ylabel('Magnitud en dB'),
grid,
zoom on
subplot(2,1,2),plot(W,fas),title('Fase del filtro'),grid on
```

ANEXO C. PROGRAMA DEL FILTRO DIGITAL EN CODEWARRIOR.

```
#include "Cpu.h"
#include "Events.h"
#include "AD1.h"
#include "S1.h"
#include "S2.h"
#include "S3.h"
#include "S4.h"
#include "Spaso.h"
/* Include shared modules, which are used for whole project */
#include "PE_Types.h"
#include "PE_Error.h"
#include "PE_Const.h"
#include "IO_Map.h"

const float COEFICIENTES[]=
{
0,26,62,103,143,174,188,176,132,54,
-56,-189,-333,-468,-573,-624,-602,-494,-296,-15,
328,699,1055,1345,1519,1532,1352,970,397,-323,
-1127,-1925,-2615,-3093,-3266,-3065,-2459,-1461,-139,1393,
2976,4423,5538,6140,6087,5297,3770,1595,-1046,-3898,
-6642,-8936,-10451,-10912,-10141,-8084,-4838,-653,4081,8866,
13141,16346,17987,17706,15331,10923,4789,-2528,-10294,-17645,
-23680,-27568,-28646,-26518,-21125,-12786,-2193,9634,21440,31855,
39549,43389,42592,36841,26367,11971,-5027,-22893,-39645,-53268,
```

-61937,-64251,-59428,-47443,-29093,-5970,19656,45056,67312,83639,
91720,90006,77952,56153,26363,-8619,-45200,-79339,-106980,-124515,
-129230,-119673,-95918,-59653,-14110,36200,85915,129370,161220,
177072,174045,151205,109794,53241,-13084,-82354,-146952,-199289,
-232660,-242060,-224863,-181267,-114457,-30441,62447,154343,234852,
294192,324332,320005,279472,204949,102604,-17876,-144153,-262444,
-358969,-421496,-440803,-411900,-334859,-215133,-63298,105791,274370,
423544,535312,594617,591209,521106,387452,200667,-22202,-258894,
-484002,-671535,-797716,-843739,-798180,-658798,-433472,-140143,
194328,536318,848470,1093357,1237418,1254819,1130832,864367,
469354,-25227,-576772,-1131855,-1630267,-2009934,-2212350,-2188018,
-1901425,-1335076,-492170,602355,1902577,3344139,4848429,6328081,
7693352,8858865,9750142,10309380,10499969,10309380,9750142,885886
5,7693352,6328081,4848429,3344139,1902577,602355,-492170,-1335076,
-1901425,-2188018,-2212350,-2009934,-1630267,-1131855,-576772,-25227,
469354,864367,1130832,1254819,1237418,10933579,848470,536318,
194328,-140143,-433472,-658798,-798180,-843739,-797716,-671535,
-484002,-258894,-22202,200667,387452,521106,591209,594617,535312,
423544,274370,105791,-63298,-215133,-334859,-411900,-440803,-421496,-
358969,-262444,-144153,-17876,102604,204949,279472,320005,324332,
294192,234852,154343,62447,-30441,-114457,-181267,-224863,-242060,-
232660,-199289,-146952,-82354,-13084,53241,109794,151205,
174045,177072,161220,129370,85915,36200,-14110,-59653,-95918,
-119673,-129230,-124515,-106980,-79339,-45200,8619,26363,56153,
77952,90006,91720,83639,67312,45056,19656,-5970,-29093,-47443,
-59428,-64251,-61937,-53268,-39645,-22893,-5027,11971,26367,36841,
42592,43389,39549,31855,21440,9634,-2193,-12786,-21125,-26518,
-28646,-27568,-23680,-17645,-10294,-2528,4789,10923,15331,
17706,17987,16346,13141,8866,4081,-653,-4838,-8084,-10141,-10912,
-10451,-8936,-6642,-3898,-1046,1595,3770,5297,6087,6140,5538,4423,

```
2976,1393,-139,-1461,-2459,-3065,-3266,-3093,-2615,-1925,
-1127,-323,397,970,1352,1532,1519,1345,1055,699,
328,-15,-296,-494,-602,-624,-573,-468,-333,-189,
-56,54,132,176,188,174,143,103,62,26,0
};
void main(void)
{float y[128],prod,suma,s[528],sum;
 int m[528];
 int i,k;
 float x[401];

  prod=0;
  suma=0;
  sum=0;

  AD1_Measure(FALSE); // Bean Measure

  S1_SetVal(); // Atenuación 1x

  setReg(ADCA_ADCR2,5); // Selección de Divisor de Reloj
  setReg(ADCA_ADCR1,4);        // Modo Triggered Sequential

  for(;;)
  {
                    prod=0;
                    suma=0;
                    sum=0;

                    clrRegBit(ADCA_ADCR1,STOP);
```

```
                /* MUESTREO */
                for(i=0;i<528;i++)
                {
                    setRegBit(ADCA_ADCR1,START); // Inicio de conversión
                    while(getRegBit(ADCA_ADSTAT,CIP)){} // Conversion in
Progress
                    m[i]=getReg(ADCA_ADRSLT0); // Vector de Muestras
                }

                /* SUMA DE COEFICIENTES */
                for(i=0;i<401;i++)
                {
                    sum=sum+COEFICIENTES[i]/100000000.0;
                }

                /* FILTRO */
                for(k=0;k<128;k++)
                {
                    suma=0;
                    for(i=0;i<401;i++)
                    {
                        prod=COEFICIENTES[400-
i]*(float)(m[i+k])/100000000.0;
                        suma=suma+prod;
                    }
                    y[k]=suma;
                }

                i=0;
```

```
   /* MUESTRAS EN VOLTIOS */
                    for(i=0;i<128;i++)
                    {
                         s[i]=(y[i]*1.48)/14888.0;   // VECTOR DE MUESTRAS EN
VOLTS
                    }

  }
}
```

ANEXO D. MANUAL DE USUARIO

MANUAL DE USUARIO

DIGISCOPE

ESPECIFICACIONES

Voltaje de Alimentación: +8V, 300mA.

Número de Canales: 1.

Voltaje máximo de la señal de entrada: ± 15V.

Frecuencia máxima de la señal de entrada: 20KHz.

Función *AUTOSET* (Ajuste Automático).

MODO DE EMPLEO

1. Energizar el prototipo *DIGISCOPE* según las especificaciones mencionadas anteriormente.

2. Para encender el dispositivo oprimir el botón *ON/OFF*.

3. Al energizar el prototipo *DIGISCOPE* se observa en la pantalla LCD la presentación del dispositivo que contiene el logo de la Universidad Industrial de Santander, el nombre del prototipo, el nombre de los autores, el grupo de investigación al cual pertenece este trabajo y la sigla E3T, que identifica a la Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones.

En el caso de que se observe en la pantalla la siguiente figura, presentada por su incorrecta inicialización se debe seguir las siguientes instrucciones:

3.1. Pulsar el botón *RESET LCD* y posteriormente regresarlo a su posición inicial.

3.2. Oprimir el botón *RESET DSP*.

4. Posteriormente se podrá observar en la pantalla LCD un marco que indica que se puede conectar la señal de entrada y que en las casillas ubicadas a la derecha aparecerán consecutivamente el valor de frecuencia, voltaje pico a pico, voltaje promedio y valor RMS.

5. Inmediatamente después de conectada la señal aparecerá en pantalla su forma de onda y sus parámetros fundamentales.

El funcionamiento óptimo del prototipo esta garantizado en el rango mencionado en las especificaciones.